Interpretable Deep Learning Approaches for Biomarker Detection from High-Dimensional Biomedical Data

Dissertation
zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)

am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

vorgelegt von

Sahar Iravani

Berlin, 2022

Erstgutachter: Prof. Dr. Tim O. F. Conrad
Zweitgutachter: Prof. Dr. Ngai-Man Cheung

Tag der Disputation: November 23, 2022

Abstract

With the advances in high-throughput data acquisition technologies, the amount of heterogeneous and complex data is constantly increasing. The application of intelligent algorithms such as deep neural networks (DNNs), which learn a hierarchy of increasingly complex features from the data, is emerging as an effective paradigm for analyzing complex datasets. In medical research, however, deep learning (DL) may suffer from overfitting due to the high dimensionality of the data. The scarcity of good quality labeled data also intensifies this issue due to the expensive and time-consuming process of providing labels and metadata by the human expert. Besides, despite the simple linear operation of core building blocks of DNNs, the hierarchical combination of these blocks may result in over-parameterization, which makes it challenging to explain their behavior. The black-box nature of these models raises a severe issue regarding the trustworthiness and reliability of deployed models, especially in high-stakes prediction applications. Therefore, to analyze high-dimensional medical data using DL models in medical settings, we need to address two important questions: 1) How to deal with the curse of dimensionality and limitation of annotated data? 2) How to improve the transparency of deep learning models through interpretability, as it potentially leads to a better understanding of the data and the deployed model?

This thesis addresses these challenges in high-throughput structured data and high-dimensional imaging data modalities. We begin our study on high-throughput structured data with the application of proteomics data analysis. We robustly learn the data representation and extract the medically relevant information using DL techniques. We develop novel data analysis based on what the DL model can learn through interpreting its predictions. This information enables getting insight into the data patterns and discovering discriminating features. We also justify the reliability of the model interpretation through comprehensive quantitative assessments. We show that the proper combination of DL techniques coupled with interpretation strategies that enable an in-depth understanding of model decisions can guide towards a reliable clinical decision support system.

Further, we study DL techniques on high-dimensional imaging data. Unlike structured data where desired features appear with slight deviations, regions of interest on medical images may appear with a large deviation on different data points. Therefore, we built our image analysis on supervised convolutional neural networks (CNN), which can handle large deviations. We investigate different CNN architectures and compare their strength. Finally, we built a robust pipeline on heterogeneous imaging data with the challenging application of human spinal vertebra detection-identification. To deal with the scarcity of data, we show how different techniques, including transfer learning, data augmentation, human-in-the-loop, and synthetic generation of data in medical settings, boost generalization.

Zusammenfassungen

Mit den Fortschritten bei den Technologien zur Datenerfassung mit hohem Durchsatz nimmt die Menge heterogener und komplexer Daten ständig zu. Die Anwendung intelligenter Algorithmen wie deep neural networks (DNNs), die eine Hierarchie zunehmend komplexer Merkmale aus den Daten lernen, entwickelt sich zu einem effektiven Paradigma für die Analyse komplexer Datensätze. In der medizinischen Forschung kann deep learning (DL) jedoch aufgrund der hohen Dimensionalität der Daten unter einer Überanpassung leiden. Der Mangel an qualitativ hochwertigen beschrifteten Daten verschärft dieses Problem noch, da die Bereitstellung von Beschriftungen und Metadaten durch einen menschlichen Experten teuer und zeitaufwändig ist. Außerdem führt die hierarchische Kombination dieser Blöcke trotz der einfachen linearen Funktionsweise der Kernbausteine von DNNs zu einer Überparametrisierung, die es schwierig macht, ihr Verhalten zu erklären. Um hochdimensionale medizinische Daten mit fortschrittlichen DL-Modellen in der Medizin zu analysieren, müssen wir daher zwei wichtige Fragen beantworten: 1) Wie kann man mit dem Fluch der Dimensionalität und der Begrenztheit der annotierten Daten umgehen? 2) Wie kann die Transparenz von DL-Modellen durch Interpretierbarkeit verbessert werden, da dies potenziell zu einem besseren Verständnis der Daten und des eingesetzten Modells führt?

Diese Arbeit befasst sich mit diesen Herausforderungen in strukturierten Hochdurchsatzdaten und hochdimensionalen Bildgebungsdatenmodalitäten. Wir beginnen unsere Studie über strukturierte Hochdurchsatzdaten mit der Anwendung der Proteomik-Datenanalyse. Wir erlernen die Datenrepräsentation auf robuste Weise und extrahieren die medizinisch relevanten Informationen mithilfe von DL-Techniken. Wir entwickeln neuartige Datenanalysen, die auf dem basieren, was das DL-Modell durch die Interpretation seiner Vorhersagen lernen kann. Diese Informationen ermöglichen einen Einblick in die Datenmuster und die Entdeckung von Unterscheidungsmerkmalen. Wir rechtfertigen auch die Zuverlässigkeit der Modellinterpretation durch umfassende quantitative Bewertungen. Wir zeigen, dass die richtige Kombination von DL-Techniken in Verbindung mit Interpretationsstrategien, die ein tiefgreifendes Verständnis der Modellentscheidungen ermöglichen, zu einem zuverlässigen klinischen Entscheidungsunterstützungssystem führen kann.

Außerdem untersuchen wir DL-Techniken für hochdimensionale Bilddaten. Im Gegensatz zu strukturierten Daten, bei denen die gewünschten Merkmale mit geringen Abweichungen auftreten, können die interessierenden Regionen auf medizinischen Bildern an verschiedenen Datenpunkten mit einer großen Abweichung auftreten. Daher haben wir unsere Bildanalyse auf überwachte convolutional neural network (CNN) aufgebaut, die mit großen Abweichungen umgehen können. Wir untersuchen verschiedene CNN-Architekturen und vergleichen ihre Stärken. Schließlich haben wir eine robuste Pipeline für heterogene Bilddaten mit der anspruchsvollen Anwendung der Erkennung und Identifizierung menschlicher Wirbel entwickelt. Um mit der Datenknappheit umzugehen, zeigen wir, wie verschiedene Techniken, einschließlich Transferlernen, Datenerweiterung, Human-in-the-Loop und synthetische Generierung von Daten im medizinischen Umfeld, die Generalisierung verbessern.

Selbstsändigkeitserklärung

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Dissertation selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht. Diese Dissertation wurde in gleicher oder ähnlicher Form noch in keinem früheren Promotionsverfahren eingereicht. Mit einer Prüfung meiner Arbeit durch ein Plagiatsprüfungsprogramm erkläre ich mich einverstanden.

Sahar Iravani
Berlin, 2022

Acknowledgements

I would like to acknowledge all the people who have influenced, supported, and encouraged me in the completion of this thesis. Getting a PhD is nothing that can be achieved, without a group of people who support you in many ways. As much as I would like to mention everyone by name and list his or her contribution, such a list would be far too long for this section.

Firstly, I would like to thank immensely my adviser Dr. Tim O. F. Conrad who gave me the opportunity to work with his group at FU and ZIB to pursue my research. Without his continuous support, availability for feedback, intellectual discussions, and guidance, this thesis would not have been possible.

Secondly, I would like to thank all my colleagues in the bioinformatics unit who have provided a motivating, intellectually stimulating, and very pleasant work environment. They helped me to release my frustration and cheered me up when necessary.

I want to express my gratitude to all people who helped to shape my path. This includes all people I have met during my studies, people I met in the scientific context, during events or conferences.

I would like to immensely thank my parents Farahnaz and Bijan for supporting me with trust and faith, and motivating me to always move forward.

Finally and most importantly, I owe the deepest gratitude to my husband Ali. He has been my pillar of support, inspiring me during moments of doubt, and has been the persistent force of enthusiasm and encouragement throughout the entire thesis.

Contents

Chapter 1

Introduction

Artificial intelligence (AI) continues to attract interest in many disciplines, including healthcare and medicine. Fueled by growing computational power, big data, and machine learning (ML) techniques, AI has shown its clinical impact in different applications, including drug discovery, medical imaging, and genomic medicine (Toh et al., 2019; Wadhwa et al., 2020). However, Employing ML technologies for high-throughput data comes with its limitations and shortcomings. Limitations include high-dimensionality and data scarcity, which degrade ML models' performance, and lack of interpretability that increases the ambiguity. Overcoming these limitations can enable ML methods to become more powerful and reliable. This thesis addresses some of these points, in particular, scarcity of data and lack of explainability using advanced ML techniques in high-throughput clinical data analysis.

1.1 Artificial Intelligence and Machine Learning

Artificial Intelligence (AI) is a general term that implies the use of a computer to model intelligent behavior with minimal human intervention (Hamet and Tremblay, 2017). The application of AI in healthcare, which is the focus of this thesis, is transforming the healthcare industry and improving outcomes (Panesar, 2019). One of the early works on AI in medicine began by focusing on expert systems (Shu et al., 2019). In expert system specific rules are captured from medical experts, and translated into computer program for knowledge processing so that it can deal with quantitative and qualitative data (Tan et al., 2016). One of the early expert systems in medicine was developed to suggest antibiotic regimens for severe bacterial infections (Shortliffe et al., 1975; Shortliffe, 2012). This approach, with almost 450 rules, was impractical at the time due to the lack of system integration into clinical pipelines. By increasing computing power, expert systems have become practical in many clinical decision support systems (CDSSs) or computer-aided diagnosis (CAD) since they directly depend on logical rules and therefore are understandable and reliable. Nevertheless, rule-based approaches fall short when dealing with complex clinical decisions, for instance, where a finite number of manual features can not describe the data representation. Besides, rule-based approaches could be costly and time-consuming for scaling the system and could be impractical when data are changing faster than the ability to write new rules continually.

ML systems, on the other hand, extract the knowledge from raw data and make

decisions on unknown events based on what has been observed from historical data. The capability of defining rules from the data enables ML to overcome expert system limitations (Akkus et al., 2019; Peiffer-Smadja et al., 2020; Buchlak et al., 2020). Besides, due to its scalability property, ML systems can be enhanced constantly through algorithm and data preparations. These capabilities and success of ML in clinical studies are drawing increased interest in CDSSs (Cabitza et al., 2017; Challen et al., 2019; Rawson et al., 2019). Among different ML techniques, deep learning (DL) has recently achieved breakthroughs in many domains (Falcone et al., 2007; Young et al., 2018; Jumper et al., 2021). DL hierarchically extracts features from data by learning them automatically, as opposed to the handcrafted feature extraction in classical ML algorithms. Although DL has had a long history, a breakthrough occurred when backpropagation learning algorithm (Werbos, 1974b) was applied to neural networks (LeCun et al., 2015). More improvement and satisfactory experimental results have been achieved by the availability of large datasets, more complex and deeper architectures, and a rapid increase in the processing power of graphics processing units (Holzinger et al., 2019; Shrestha and Mahmood, 2019). DL has been proven one of the most successful ML algorithms in image recognition (Krizhevsky et al., 2012; Huang et al., 2016b), speech recognition (Hinton et al., 2012a), natural language processing (Sutskever et al., 2014), etc. This potential has also been shown in many bioinformatics applications, including biomedical image processing and diagnosis (Esteva et al., 2017; Kermany et al., 2018), biomedical signal processing (Rashid et al., 2020), biomolecule interaction prediction, and systems biology (Ma et al., 2018b; Zitnik et al., 2018; Piccialli et al., 2021). In the following sections, we discuss how healthcare systems can benefit from ML/DL models. We further explain the challenges towards the integration of these models into healthcare systems like CDSS.

1.2 Clinical Decision Support System

As briefly introduced earlier, CDSSs are computer systems initiated to deliver better healthcare through providing clinicians, staff, patients, or other individuals with targeted clinical knowledge, patient information, and other health information (Osheroff et al., 2012). CDSSs are more established as systems to help clinicians as a second opinion by delivering a variety of services, including diagnosis, prognoses, treatment

response prediction, treatment recommendation (personalization), and many more. Traditional CDSSs are traced back to 1970s (Shortliffe and Buchanan, 1975). The poor system integration, ethical and legal issues, and imperfect explainability at the time had limited the applicability of these systems to academic pursuits (Middleton et al., 2016; Shortliffe and Buchanan, 1975). Today's CDSSs take advantage of web applications, integration with electronic health records, or computerized provider order entry systems, which can be administrated through desktop, tablet, smartphone, and other devices such as biometric monitoring and wearable health technology (Dias and Paulo Silva Cunha, 2018). CDSS provides a variety of functions, including improving clinical workflow, patient safety, quality of care, healthcare efficiency and disease management, and diagnostic assistant tools (Berner, 2007; Shahsavarani et al., 2015; Sutton et al., 2020). For patient safety and quality of care, CDSSa are greatly enhanced towards reducing prescribing and dosing errors (Helmons et al., 2015), contraindications through automated warnings (Peris-Lopez et al., 2011), and drug control events (Jia et al., 2016). Towards healthcare efficiency, the CDSS's function is extending to reduce the health system cost containment(Calloway et al., 2013), and administration function, which directly helps clinical protocols (McEvoy et al., 2018). CDSSs have also been developed and applied across a variety of diagnosis systems, including infectious diseases (Shen et al., 2018), Alzheimer's disease (Toro et al., 2012), skin cancer (Curry and Reed, 2011), breast cancer (Mazo et al., 2020), imaging diagnosis (Curry and Reed, 2011), and chronic disease grading (Nejati et al., 2016).

Based on how CDSSs are derived, they can be classified into knowledge-based and non-knowledge-based systems. Knowledge-based CDSSs are expert-driven systems in which the rules are made using literature-based, practice-based, or patient-directed evidence (Sim et al., 2001). Non-knowledge-based CDSSs rather leverages ML models. ML extracts knowledge from the historical clinical data and builds a predictive model to estimate the outcome of new observations (Berner, 2007; Cabitza et al., 2017; Challen et al., 2019; Rawson et al., 2019). The outcomes are then used as a recommendation system to support clinicians in their practice. There is a great interest in non-knowledge-based CDSSs using ML algorithms to enhance clinical decisions' accuracy and minimize medical errors, e.g, in diagnosis system for detection of diabetic retinopathy in retinal fundus photographs (Abràmoff et al., 2018) and infection disease (Lamping et al., 2018), and in improving clinical workflows for analysis of imaging modality (Akkus et al., 2019)). Despite advantages, there are still

challenges that should be addressed for adopting ML/DL models into complex medical data analysis such as high-throughput data and integration of these models into CDSS. These challenges include: 1) providing good quality historical data (which is usually scars) to train a robust ML model, and 2) explaining the outcomes of the black-box nature of ML models to bring trust to the users. In the following sections, we discuss these shortcomings in the context of high-throughput data analysis. We first introduce high-throughput data analysis and then elaborate on the challenges.

1.3 High-throughput Data Analysis

High-throughput data are information generated in massive and have the potential to improve our understanding of the biological system significantly (Porter and Hajibabaei, 2018). The term high-throughput in the literature is used when the number of observations, number of features, or both are gigantic. High-throughput technologies enable comprehensive study of biological processes by measuring multiple parts of a biological system, simultaneously and at the reduced cost. Therefore, they are widely used in modern medicine, and diagnostic systems (Eicher et al., 2020; Ristevski and Chen, 2018; Viceconti et al., 2015). The rapid development of high-throughput quantification tools for different modalities, (e.g., omics data (Guan, 2015; Albaradei et al., 2021), high-throughput imaging data (Huizing et al., 2019), medical health records data (Yu et al., 2015; Smoller, 2018), and sensor arrays) and the variety of research questions are constantly increasing its popularity in different fields, including automated diagnosis, prognosis, and drug design.

We refer to a dataset as high-throughput only when the number of features greatly outnumbers the number of observations. Unfortunately, this phenomenon that is called the curse of dimensionality makes the high-throughput data analysis prone to overfitting for classical ML methods. The scarcity of annotated data is mainly due to the expensive and time-consuming process of providing annotation by the human expert. In addition, noise content in such data is often high as a result of several stages of data acquisition. These characteristics make analyses of high-throughput data challenging for ML researchers, and slow their integration into clinical workflows.

This thesis elaborates on the challenges mentioned above and presents possible solutions through developing new ML/DL techniques based on recent advances in

this field, which potentially can pave the way for more accurate diagnosis. We adopt ML models to analyze two types of high-throughput data: 1) Mass spectrometry proteomics data that has emerged as a standard tool for large-scale protein analysis of biological samples (e.g., blood), and 2) Magnetic resonance imaging data that has been established as non-invasive diagnostic tool for analyzing internal body structures. These data are characterized by few observations, lots of features or high complex content, and noise interference.

1.4 Challenges and Contributions

As previously stated, the demand for AI using ML and DL is increasing in the field of health informatics. Their potential benefits have already been shown in many applications, such as, supporting clinicians to diagnose diseases (Peiffer-Smadja et al., 2020), identifying cancer biomarkers (Sharma and Rani, 2021), and predicting outbreaks of infectious diseases (Khakharia et al., 2021). Despite conventional methods, the DL approach does not require domain-specific data preprocessing, and it is expected that it will enable the healthcare assistant tools towards full automation in the future (Nakkiran et al., 2019). In spite of all the advantages, there are still challenges to adopting ML/DL models for high-throughput data interpretation.

The first challenge is related to data scarcity. In medical applications, data acquisition and ground truth annotations are often very expensive and time-consuming. Because it requires a considerable amount of time from human experts that could be in return dedicated to patients. Sometimes, annotations could even be infeasible to acquire due to millions of features that are needed to be investigated. High dimensionality and scarcity of data, which is also known as the curse of dimensionality, increase sparsity. To assure that the model remains valid, the amount of needed data grows exponentially. To address this problem, ML models are conventionally equipped with a dimension reduction preprocessing step (Meng et al., 2016) that removes the irrelevant and redundant features, and a feature selection method that chooses the best subset of features.

The feature selection methods for sparse high-dimensional data can be divided into three categories: filter, wrapper, and embedded methods (Chandrashekar and Sahin, 2014; Espadoto et al., 2019). Filter methods select features based on general characteristics of training data, e.g., Fisher's t-test, an information-theoretic criteria.

Filter methods evaluate the importance of each feature in univariate or multivariable measures, independent of the learning model. The wrapper methods (e.g., simulated annealing or genetic algorithms) use the searching techniques to select feature subsets and a learning algorithm to evaluate these subsets in terms of classification error or accuracy (Dash and Liu, 1997). Compared to filter methods, the wrapper methods achieve better feature subsets to enhance the performance of a predefined learning algorithm. The wrapper methods, however, are more computationally demanding, which makes their application to high-throughput data analysis infeasible. Similar to wrapper methods, the embedded methods, e.g., LASSO, SVM, and ElasticNet, also select features based on the learning algorithm. However, unlike the wrapper method, they simultaneously select the most important features during the training phase. Although the embedded methods are shown to be computationally more efficient, they rarely reach better learning performance than wrapper methods (Hancer et al., 2020). In summary, adopting conventional feature selection to the application of high-throughput data is either computationally inefficient (or even infeasible) or affected by overfitting and feature-biased problems. As for medical data, conventional features selection methods can raise the risk of losing relevant biological information. This thesis proposes ways to analyze such data in their raw format by developing new models based on DL techniques capable of handling large data due to their scalable characteristics. We exemplify our analysis in different biomedical data modalities and extract the biologically relevant information through the means of model interpretation.

To overcome the scarcity of data and avoid overfitting, we rely on the fact that the neural network can memorize complex features and has scalable capacity to learn these complex patterns. Therefore, the key to obtaining a proper generalization performance is to allow the network to be exposed to more variants of the data. These variations on high-throughput structured data are modeled through synthetic generation of data in Chapters 3 and 4. To achieve better generalization performance, we demonstrate how one can effectively benefit from transferring the knowledge from datasets that share similar representations. On structured proteomics data, variations are synthesized through data simulation in Chapters 3 and 4. On imaging modality, variations are modeled through proper data augmentation in Chapter 5. We further investigate enriching the annotated data by the semi-supervised generation of labels with the help of human interference in Chapter 5.

The second concern surrounding ML/DL in medicine is the lack of model trans-

parency (Topol, 2019). As decisions made or influenced by these systems affect human health, it is crucial to understand how and why a system has produced a given output (Arrieta et al., 2020). Therefore, ML-based CDSS is in great need of explainable AI, which can explain how recommendations are made. This can allow practitioners to decide when and where to trust the model, especially in high-stakes cases. Lacking proper interpretation that is understandable by the users in ML/DL models, can be one of the reasons that still less complex models, for instance, rule-based expert systems, have preferably been used in CDSS. Rules are easy to interpret, and therefore they can be used reliably. But, they may not be capable of solving complex tasks, where thousands of features in the data should be evaluated for making decisions. In this thesis, to convey a sense of trust to the users and demonstrate the model's reliability and trustworthiness, we employ the new advances of interpretation strategies that enable understanding the black-box nature of DNNs. In Chapter 3, we show how to adopt these strategies into our DL-based high-throughput data analysis.

Model interpretations can also be translated into building units to enhance the model's generalization, find possible bugs, and provide new insight into the analyzed data. Towards these goals, we quantitatively measure and visually assess the interpretation of decisions made by the network in Chapter 4. These assessments provide systematic information regarding the model behavior and patterns in the data. We show how to utilize this information from interpretation assessments to enhance the models that is designed for high-dimensional data analysis, and to reveal relevant discriminating patterns of data.

1.5 Thesis Overview

This thesis aims to address challenges that are essential for successfully applying DL in high-dimensional biomedical data, which provides tools and in-depth insights towards automatic healthcare as a support for physicians. This chapter has described the general motivation of this thesis by highlighting the importance of AI in medical settings and the challenges of analyzing high-throughput data for adopting modern ML approaches. Chapter 2 gives an introduction to the data modalities. In Chapters 3, 4, and 5 the analyses listed below will be investigated:

Analysis of high-dimensional structured data using modern machine learning techniques towards finding objective indications of medical states: In Chapter 3, we shed light on difficulties of analyzing high-dimensional structured data, including the high-dimensionality, scarcity of good quality labeled data, and high noise level. These characteristics pose challenges to making a diagnosis, extracting relevant information, and finding relevant patterns associated with medical states. This task so far has been addressed either with statistical analysis, which is often not accurate enough, or with ML approaches, which can be prone to overfitting. Moreover, ML methods may suffer from lack of interpretability, and are limited to the capacity of the model. We propose an interpretable DL-based approach for classifying such data and identifying biomarker candidates to address these challenges. We successfully apply our proposed method on simulated and real-world datasets.

Understand and enhance developed machine learning models through interpretability assessment: In Chapter 4, through interpretability assessments we demonstrate how to understand the behavior and enhance the architecture of high-throughput data predictive models. The interpretation of a model gives an insight into the reasons why the model makes certain decisions. These reasons can be quantified and act as feedback to modify model parameters or serve as an evidence of the model's robustness. We develop a quantitative interpretation assessment of DNN predictions to tune the DL model parameters and adopt visualization assessments to understand the model's behavior.

Translate acquired knowledge into analysis of medical imaging modality: In Chapter 5, our knowledge from analyzing high-throughput structured proteomics data will be transferred into high-dimensional medical imaging data. We target analyzing challenging high-dimensional magnetic resonance imaging (MRI) data of the human anatomy to detect and identify the region of interest. Besides the complexity itself, these tasks also suffer from high-dimensionality and a small labeled sample size (similar to structured proteomics data analysis in Chapters 3 and 4). Unlike the structured data the regions of interest (ROI) on imaging modality can be appeared in different parts of the data. It means the position ROI largely varies from one data points to another. Chapter 5 adopt convolutional neural networks to handle spatial variations in imaging modality, which require minimal preprocessing.

We compare different supervised DL architectures (e.g., UNet (Ronneberger et al., 2015) for Segmentation, YOLO (Redmon et al., 2016) for detection, and ResNet (He et al., 2016a) for Regression) and propose a robust pipeline for detecting regions of interest on heterogeneous imaging data. This study can add valuable insight into the image data analysis and open new ways to solve medical imaging tasks.

In summary, we successfully build robust DL classifiers on raw high-throughput mass-spectrometry proteomics data to identify the medical states of proteomics samples. The scarcity of data is coped with transfer learning and leveraging synthetically generated data. We then propose a novel biomarker detection using the robust classifier and the means of interpretation. Compared to existing methods, we find a more accurate subset of biomarker candidates and reduce many otherwise necessary preprocessing steps, which lessen the dependency on human-expert knowledge. We quantitatively assess the interpretation of developed models to demonstrate reliability and achieve better configurations. In addition, with the comprehensive study of the high-dimensional biomedical imaging analysis, we successfully propose workflows for segmentation, detection, and regression of regions of interest on such data. Our developed methods have been actively used by a project partner for interpretation and annotations of MRI imaging data.

Chapter 2

A Primer on High-throughput Data Modalities

Before we jump into the data analysis, we will introduce the data modalities used in this thesis, including 1D and 2D high-throughput mass-spectrometry proteomics and human organ high-throughput medical imaging.

2.1 Proteomics

Proteomics is the large-scale study of proteins that are regarded as vital parts of a living organism and functional units of cells. Proteins control all biological systems in a cell and regulate the body's tissues and organs. Therefore, identifying the presence of proteins and their alterations is required to develop diagnosis systems. Towards this goal, mass spectrometry has evolved into an indispensable tool for the comprehensive identification of all proteins and their abundance levels in complex analytes. To explain why proteomics is investigated in general and how it differs from genomics and transcriptomics, we will have a gentle introduction in the following.

2.1.1 DNA, RNA, and Protein

Genomics is the study of the genome or (mainly) the entire set of DNA (deoxyribonucleic acid) that contains genetic information for the development and functioning of cells and organisms. DNA is made of two strands that wind around each other by paring four chemical compounds known as nucleotide bases. The four bases are adenine (A), tyrosine (T), guanine (G), and cytosine (C). The bases pair up with each other - A with T and C with G - between the two strands and form a spiraling ladder shape, known as a double helix shape. Three consecutive nucleotides, termed a codon, form a unit of genomic information (e.g., ACT, CAG, TTT) and encode for a specific amino acid (or stop signal). The sequence of codons then encodes for the sequence of amino acids, which form proteins. Four nucleotide bases can happen in $4^3 = 64$ different codons, enough to encode for the 20 existing amino acids. The flow of information from DNA to RNA and then from RNA to proteins consists of two major steps: transcription and translation. Through the process of transcription, the codons of a gene are copied and formed into an RNA molecule (called messenger RNA) in the nucleus. After the RNA molecule leaves the nucleus and enters the ribosome, the sequence of codons is read and translated into a chain of amino acids. Proteins are typically composed of hundreds of amino acids in sequence, held together by chemical bonds. Different combinations of amino acids give

proteins different shapes, sizes, and functions. Besides, proteins may become further altered with post-translational modifications in their lifetime for better-specialized functions.

RNA (Ribonucleic acid) presented in all biological cells is a single-stranded nucleic acid with base uracil (U) instead of base T. While DNA contains all the instructions for the cell to grow, function, and replicate, RNA carries out these instructions. It copies and transfers the genetic code from the DNA to make relevant proteins. Cells contain many types of RNA, including messenger RNA and transfer RNA. Messenger RNA evolves in the translation of genes from DNA into proteins, while transfer RNA serves as the link between the messengers RNA and the amino acid sequence of a protein.

During the translation process, the molecular complexity increases by several mechanisms, generating from an estimated 20,000 human genes (Kuster et al., 2005) to between 70,000 and 100,000 transcripts with varying abundance levels (Gaudet et al., 2017). These transcripts are converted into even a larger number of different protein sequences because of sequence mutations, alternative translations, and post-translation modification. Further, the enormous variation in protein abundance and its concentration over time expand the number of different protein products even more. These alterations sometimes cannot be captured directly from DNA or RNA, where proteomics fills the gap.

Changes in the abundance of proteins can influence the state of an organism. One focus of proteomics, for instance, is to study the changes in protein abundances generated responding to a perturbation, disease, morphogenesis, toxicity, or other cell stress in a given biological system (Villanueva et al., 2006; Faca et al., 2009; Calvo et al., 2009; Magni et al., 2010). All efforts to assign these responses to changes in proteome rely on the identification and quantification of proteins that are presented in a sample. The success of mass spectrometry (MS)-based proteomics illustrates its role as an indispensable tool towards the goal of Proteomics research, which is the comprehensive identification of proteins, their abundances, and concentrations presented in an analyte. It does not require prior information on which proteins are presented in a sample, and it allows identification and quantification of thousands of analytes in parallel. MS-based approaches have contributed significantly to our understanding of life by comprehensively mapping entire proteomes.

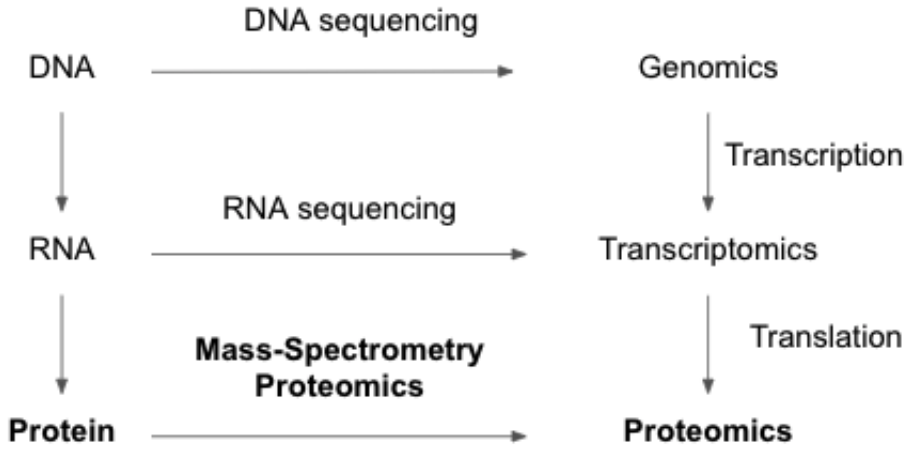


Figure 2.1: Overview of a hierarchy of omics approaches: genomics, transcriptomics, and proteomics. Each level of information in DNA, RNA, and proteins in these approaches provides a different levels of characterization of the systems biology. The flow of genetic information within a biological system is studied through genomics, transcriptomics, and proteomics. Copying the DNA information into RNAs is referred to as transcription, and that by which RNA is used to produce proteins is called translation. Proteomics is the large-scale study of proteins, which is investigated by mass spectrometry analysis in Chapter 3.

2.1.2 Mass-Spectrometry Proteomics

Mass spectrometry (MS) is a high-throughput data acquisition technique that can be used for study of proteins. In this analytical technique, chemical compounds are first ionized into charged molecules, and then their mass to charge (m/z) ratio are measured. A mass spectrometer is made of three parts: ionizer, mass analyzer, and detector. The ionizer or ion source transfers the analyte, e.g., peptides, to the gas phase and ionizes the gaseous analyte afterwards. Although MS was discovered in the early 1900s, the development of matrix-assisted laser desorption ionization (MALDI) (Karas and Hillenkamp, 1988), and electrospray ionization techniques (ESI) (Fenn et al., 1989) in 1980s, increased the applicability of MS to large biological molecules like proteins (Singhal et al., 2015). Both ionization techniques convert the peptides into ions by adding or losing one or more than one protons. These techniques are also known as soft ionization, which do not cause major fragmentation. The datasets that are used in this thesis are generated by these ionization techniques, which will be introduced later in this section.

The mass analyzer receives the ionized analytes and separates the charged particles based on their mass to charge (m/z) ratio. There are different types of mass analyzer, including magnetic sector, time-of-fligh (TOF), quadrupole, and ion trap. The detector then detects a signal produced by the charged ions. For instance, in TOF, the signal is the time that accelerated ions take to travel a distance in an electric field (Mamyrin, 2001). In ion trap, the signal is measured by the oscillation frequencies along an electrode (Makarov, 2000). The recorded signal in this setting is known as mass spectrum, MS scan, or MS1 scan. The mass spectrum is a plot that has mass-to-charge ratio (m/z) on the x-axis and intensity values – ion count – on the y-axis. The masses are measured in the Dalton (Da) or unified atomic mass unit (u).

A limitation of the MS1 scan or 1D MS data is that the gained information cannot determine the exact sequence of peptides whose ions have very similar m/z ratios. To address this limitation, an extra mass analyzer can be added to the mass spectrometer, which helps to distinguish the ions with close m/z ratios. To this end, ions with a particular m/z from the first analyzer are isolated and fragmented into smaller ions. Some commonly used fragmentation methods include collision-induced dissociation (CID)(Wells and McLuckey, 2005), higher energy collision-induced dissociation (HCD) (Olsen et al., 2007), and electron-transfer dissociation (ETD) (Syka et al., 2004). CID collides the ions into fragments using neutral gas molecules, e.g., nitrogen or argon gas. In HCD, a high voltage breaks the ions with the gas. ETD induces ions' fragmentation by transferring electrons to them. Once the ions are fragmented, the second MS analyzer receives and measures the m/z ratio of the fragmented ions. The signal from the second scanner is termed MS2 scan, which enables identifying the analyzed peptides because the masses of all amino acids and most common post-translational modifications are known.

Peptides from MS2 scans can be identified by different approaches, such as *de novo* (Taylor and Johnson, 1997) and spectrum library search (Frewen et al., 2006). *De novo* tests the combination of amino acids and checks if their masses are equivalent to the fragmented ions. Spectrum library search matches the derived spectra against existing identified peptides. While spectrum library search approach uses an extensive search, the database search engine uses protein databases to match spectra to peptides. Databases contain amino acid sequences of all known human proteins, such as the UniProt database (Bairoch et al., 2005), and the National Center for Biotechnology Information, (Wheeler et al., 2007).

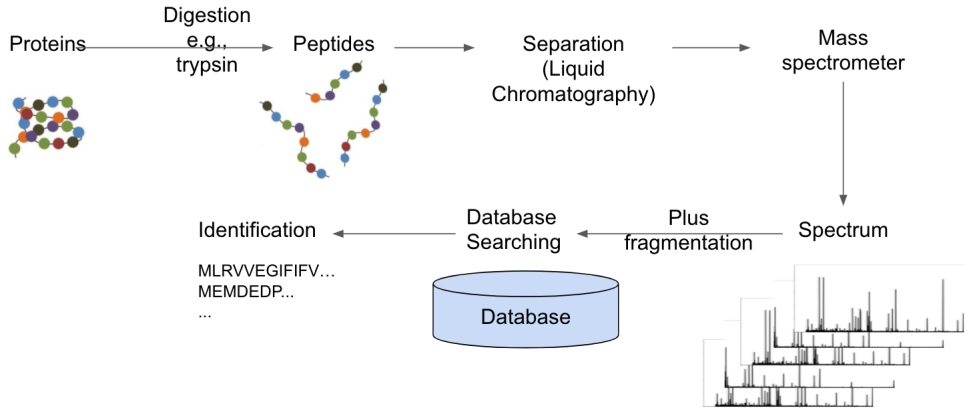


Figure 2.2: Overview of a bottom-up proteomics workflow. In the proteomics study, a protein mixture is typically digested to break the proteins into smaller units of peptides, which are then further separated with the means of liquid chromatography and mass spectrometry. Finally, the generated spectrum derived from a fragmentation is used for protein identification through database searching.

Mass spectrometry proteomics studies proteins in bottom-up or top-down approaches. Top-down proteomics measures all modifications that occur on the same molecule, which enables the identification of precise proteomes. However, in bottom-up proteomics, protein or peptide mixtures are first subjected to enzymatic cleavage. This process generates peptides that are more tractable to experiment and compute, making bottom-up proteomics the most widespread workflow (Dupree et al., 2020). Figure 2.2 illustrates an overview of a bottom-up proteomics workflow. In typical bottom-up proteomics approach, a sequence-specific enzyme, e.g., Trypsin, digests proteins. The resulting peptide products are analyzed using a mass spectrometer coupled with high-performance liquid chromatography (HPLC) system or MALDI-TOF instruments. Next, a database searching approach is used for protein identification. The following sections introduce 1D an 2D mass-spectrometry proteomics datasets that are used in this thesis using MALDI-TOF and HPLC-ESI instruments.

2.1.3 1D Mass Spectrometry Proteomics

We refer to MS data as 1D when the MS analyzer generates one spectrum per sample. As previously stated, there are different ionization techniques and analyzers that can be used for mass spectrometry. MALDI-TOF MS is one of the most popular MS techniques (Hou et al., 2019), which is also used in this thesis as 1D MS proteomics

data. In MALDI-TOF MS, the molecules of the examined sample are vaporized, ionized, and finally analyzed by their respective TOF through an electric field. MALDI stands for Matrix-assisted laser desorption ionization. To facilitate the ionization process, this approach uses a small organic molecule called Matrix. The Matrix comprises Benzoic acid and cinnamic acid. The mixture of the Matrix and analyte is deposited onto the sample plate, known as the target. Adding the Matrix allows solvent to evaporate, solidify and trap the sample. The solid sample on the target is then analyzed by MALDI-TOF spectrometer. The target is vacuumed and hit by a laser beam. The Matrix absorbs the laser energy and provides this energy to ablate from the surface of the samples, and carry the analyte molecule into the gas phase. The molecules are ionized by a proton transfer during the ablation process, often with a single charge ($z = 1$). Note that the analyte itself would be destroyed by hitting the laser in the absence of the Matrix. Now, the masses of the gas-phase ions can be analyzed by the TOF MS. The TOF process accelerates the molecular ions in a high voltage electric field. This filed makes constant kinetic energy: $E_{kin} = \frac{1}{2}mv^2$, where m and v denote the ion mass and velocity, respectively. At a fixed kinetic energy, ions with different m/z ratios are accelerated to different velocity, which is inversely proportional to their mass so that small ions reach the detector first. The detector records the time that the ions spend travelling a certain distance. Mass spectrometer calculates the m/z of ions using the measured time and constant kinetic energy. Then, it forms a mass spectrum with mass to charge ratio of ions on the x-axis and the ion intensities on the y-axis. The scale on y-axis generally contains more than 10000 m/z ratios.

MALDI is considered a soft ionization technique. It means the molecules are turned into the gas phase without fragmentation, and they remain intact when they are ionized. The soft ionization allows the direct mass spectrometry of mixed samples. However, it is difficult with this process to identify larger micro molecules, e.g., peptides, that have overlapping masses. In such cases, for more precise and accurate identification, for instance, ESI-MS can be used, which is readily coupled with the chromatography separation technique. This technique allows online separation during MS analysis and, thus, is more widely used for complex mixtures of proteins.

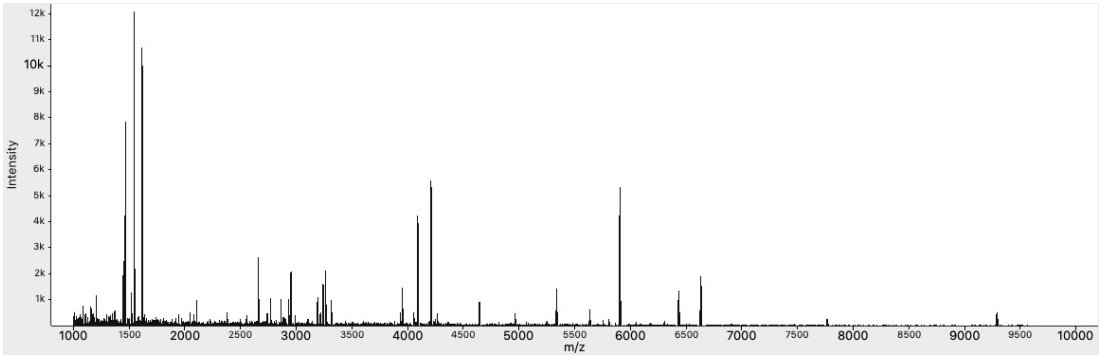


Figure 2.3: MALDI-TOF MS proteomics sample visualization in OpenMS (Röst et al., 2016). The x-axis shows the mass-to-charge ratio (m/z) of the ions captured by the mass spectrometer, and the y-axis demonstrates their intensities. This is the input vector for the MALDI-MS analysis in Chapter 3.

2.1.4 2D Mass Spectrometry Proteomics

Liquid Chromatography

The complexity of proteomics samples may require a separation step prior to the mass spectrometry analysis to reduce the complexity of peptide identification. This separation leads the analyzer to generate separate spectra. We refer the generated data as 2D MS data. One of the most common peptide separation techniques is high-performance liquid chromatography (HPLC) (Liu et al., 2020b). HPLC consists of a solvent, and a chromatography column termed mobile and stationary phases, respectively. In HPLC, the peptide solute is first injected into the chromatography column. This solute is then forced through the column at high pressure, which effectively separates the components based on the chemical affinity (e.g., hydrophobicity, ionic interactions) and weight. To run the solute through the column, molecules of the sample require a different amount of time, so-called retention time (RT). RT measures the time from when the solvent injected into the column until the components elute from the column. The difference in RT separates the peptides of different species. The components that leaves the chromatography column are in the form of droplets, which need to turn into gaseous ionized peptides prior to mass spectrometry.

ESI

ESI is one of the ionization techniques that can be coupled online to the HPLC. ESI produces peptide ions with different charges and allows analyzing compounds in the solution. In this technique, analyte solution is forced through a needle at the tip of the LC column. A high voltage is applied to the needle, which forms charged droplets of peptide solution to be entered to the ESI source. In ESI, the droplets are directed through the heated desolvation region, which evaporates the solvent and turns the charged peptides to the gas phase. The desolvated peptide ions are then entered into the mass spectrometer. Similar to MLDI, ESI is also considered a soft ionization technique, as it leaves the analyte mostly intact.

LC-MS Map

Stacking the spectra derived from the HPLC at a specific interval forms a liquid chromatography-mass spectrometry (LC-MS) dataset. Retention time adds a dimension to the individual mass spectrum and builds a three-dimensional map whose axes represent retention time, m/z ratio, and ion count (intensity), respectively. An examples of LC-MS map is shown in Figure 2.4. LC-MS data is considered a 2D MS data analyzed in this thesis.

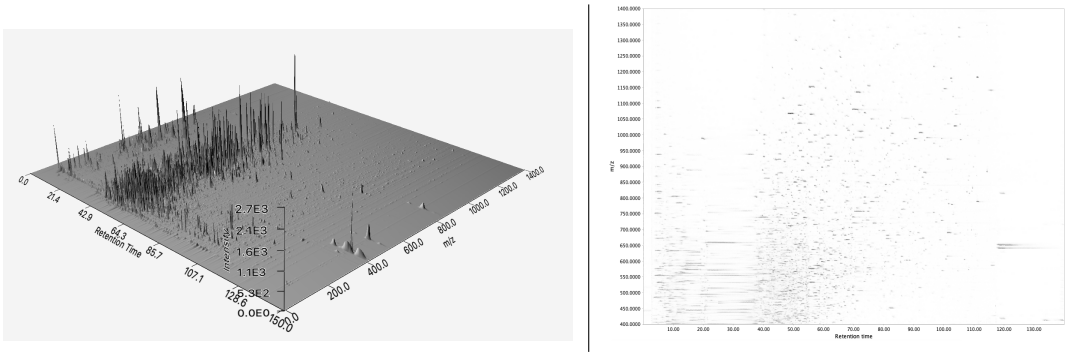


Figure 2.4: Visualization of an LC-MS map in MZmine 2 (Pluskal et al., 2010). Left panel shows a 3D visualization of an LC-MS map, where x, y, and z axes demonstrate mass-to-charge ratio (m/z), the retention time, and ion intensities, respectively. Right panel is a 2D visualization of the same sample from the top perspective. The gray-scale color values demonstrate the ion intensities. This image is the input to our LC-MS data analysis pipeline in Chapter 3.

2.2 Medical Imaging

Medical imaging has delivered rich information about the human anatomy and is usually the initial source of investigation for various diagnoses. There are different imaging modalities, including computed tomographic scanning (CT) and magnetic resonance imaging (MRI), which are used for different purposes. In comparison to CT, MRI has the advantages of improved contrast resolution for bone and soft tissues, along with the versatility of direct imaging in multiple planes. These imaging techniques play an essential role in medical imaging diagnoses. This thesis focuses on spine MRI interpretation, including vertebra segmentation, localization, and identification, which assist clinicians in their everyday workflows. We will expand this application in detail in Chapter 5. The following section gives a gentle introduction to this clinical imaging modality and the anatomy of the spine itself.

2.2.1 MRI

MRI produces detailed anatomical images in axial, sagittal, and coronal dimensions. It is a non-invasive imaging technology for diagnosing a broad range of soft-tissue conditions. MRI does not use x-rays or other radiation, which makes it the imaging modality of choice for diagnosis, treatment, and monitoring, especially when frequent imaging is required.

MRI relies on the magnetic properties of hydrogen atoms, which are abundant in the human body. It uses a strong magnetic field of 1.5 or 3 Tesla and radio waves, which excites and captures the energy released by changes in the direction of the rotational axis of protons in the body tissues. Hydrogen protons, which act like a tiny magnet, are all in random positions and spinning on their axes. Therefore, there is no magnetic field. The spinning motion is known as precession. A strong primary magnetic field in MRI first aligns the protons with the field and affects how fast these protons spin. Next, the gradient coil in MRI generates a secondary magnetic field within the bore of the primary magnet that varies across the body. The arrangement of the gradient coils enables MRI to image along the X, Y, and Z axes. This happens by varying the strength of the primary magnetic field, which changes the precession frequencies between slices, allowing spatial encoding for MRI images. The X, Y, and Z gradients run along horizontal, vertical, and long axes to generate sagittal, coronal, and axial images.

A radiofrequency pulse is then introduced from radiofrequency coils, which turns some low-energy protons into a high-energy state and flips them away from the primary magnetic field. The radiofrequency should be the same as the frequency of the spinning hydrogen protons so that the protons can absorb energy and rotate. Radiofrequency coils are designed for specific body regions.

Once the radiofrequency pulse is turned off, the protons flip back and realign along the primary magnetic field. This realignment releases electric-magnetic field energy captured by the MRI sensors. Different tissues release different amounts of energy in different periods of time. Therefore, various tissues can be differentiated based on the amount of the released energy and the time it takes for the protons to realign with the magnetic field since the radiofrequency pulse has turned off. The Changes in the released magnetic field along the way generate electric current. A Computer receives an analog electrical signal and converts it to a digital signal. Then, the digital signals are transformed into MRI images using Fourier transformation. For more details on physics of MRI, please see (Plewes and Kucharczyk, 2012).

2.2.2 Spinal Anatomy

The human spinal column, also known as the vertebrae column, contains 24 vertebrae and two sections of naturally fused vertebrae – the sacrum and the coccyx. The vertebrae column is divided into five sequences of regions stacked on top of each other: cervical vertebrae in the neck, thoracic vertebrae in the upper back, lumbar vertebrae in the lower back, the sacral vertebra in the sacrum, and coccygeal bones located below the sacrum. In this thesis, we refer to the first three regions as the vertebrae column. There are seven cervical (C_1-C_7), 12 thoracic (T_1-T_{12}), and five lumbar (L_1-L_5) vertebrae, as it is illustrated in Figure 2.5.

Each vertebra comprises a cylinder-shaped bone in front of the spine called the vertebral body, separated by interval discs and paired facet joints in the back. All vertebrae are knitted together by ligaments and tendons in an S-shaped curve. They support and stabilize the spine while allowing balance maintenance, shock absorption, and mobility.

Spinal anatomy is visualized by different imaging modalities such as CT, X-ray, and MRI. While CT and X-ray may subject patients to ionizing radiation, limiting the number of scans that can be taken from patients, MRI is considered a non-invasive and safe modality uses of non-ionizing radiation (Watson, 2015). Spine

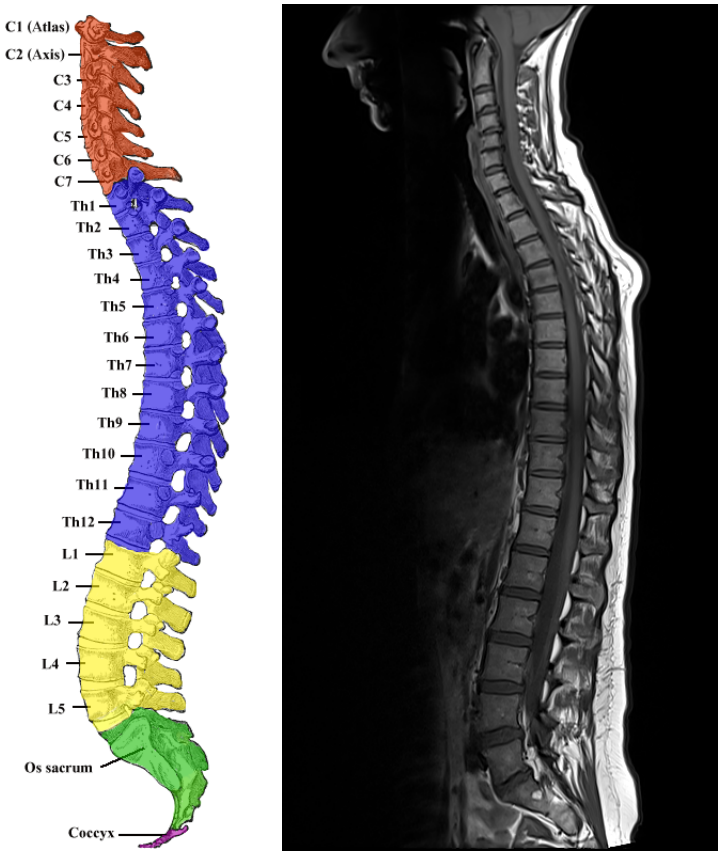


Figure 2.5: Lateral view of the human spine. Left panel: Spinal vertebrae anatomy (taken from Henry Vandyke Carter, Public domain, via Wikimedia Commons, 1918). Right panel: Spinal vertebrae MRI scan. The spinal column consists of 24 individual vertebrae, including seven cervical (C_1-C_7), 12 thoracic (T_1-T_{12}), and five lumbar vertebrae (L_1-L_5). Spinal column disorders can be diagnosed through MRI scans by showing the shape and structure of bones, discs, spinal cord, and spaces between the vertebral bones. The last five vertebrae, the lumbar region, support most of the weight of the human body. This region causes the pain for the majority of people. Many other spinal diseases are also associated to lumbar region. Robust localization and identification of this region is an essential and primary step for diagnose of various spinal disorders, which is addressed in Chapter 5.

MRI scans show the bones, discs, spinal cord, and the spaces between the vertebral bones. Therefore, they can be used in a wide variety of spine-related assessments, including the spine alignment; trauma injury to the bone, disc, ligament, or spinal cord; spinal cord inflammation or compression; disc and joint disease; and tumors in vertebrae and their surrounding soft tissues.

Chapter 3

A Novel Interpretable Deep Learning Feature Selection Approach for High-throughput Omics Data

High-throughput biomedical technologies create large datasets that routinely need to be interpreted in medical settings to investigate the medical relevant patterns. This chapter presents a novel biomarker detection approach for high-throughput mass spectrometry proteomics data, which employs modern machine learning techniques. We build a robust DL classifier on complex proteomics data with a limited number of training instances through transfer learning. Then, we employ the trained classifier and explain its decision using a proper interpretation method to extract biomarker-relevant information.

The produced data through high-throughput biomedical techniques results in degrees of complexity. This complexity is often caused by the high-dimensionality, scarcity of good quality labeled data, and high noise level. Besides, integration of any analysis into the medical decision support system requires decision interpretation for trustworthiness and reliability, which are hard to acquire with complex algorithms. This chapter sheds light on these difficulties and studies the application of modern ML/DL techniques to resolve them. First, we investigate new advances in deep learning interpretation methods and show what we can learn from these interpretations. We then propose an interpretable deep learning model to extract relevant information from high-throughput data. Our framework learns the representation of instances by training the deep neural network, and realizes the relevant patterns based on what the machine has learned. We formulate this problem in the context of high-throughput proteomics data classification and biomarker detection with the case studies of MALDI-MS and LC-MS data analyses, introduced in Sections 2.1.3 and 2.1.4.

To set the stage, we will first briefly overview the background of the DL approach in Section 3.1, the background of various explanation methods in Section 3.2, and the background of deep transfer learning for generalization purposes in Section 3.3. Section 3.4 reviews the related work on high-throughput proteomics data analysis and the challenges that DL and ML models pose in this analysis. In Section 3.5 and 3.6, our proposed method for MALDI-TOF MS accompanied with an evaluation study will be presented. Then, we will extend our feature selection approach utilized in the proposed DL pipeline to more complex data, LC-MS data, in Section 3.7. The evaluation of this extension is demonstrated in Section 3.8. We will finally discuss the main findings and limitations in Section 3.9.

3.1 Deep Learning

Deep learning (DL) is a subset of machine learning algorithms, which attempts to learn high-level abstractions in data by hierarchically extracting features. Although DL has had a long history, it has just recently achieved more satisfactory experimental results. One of the early prototypes developed by McCulloch and Pitts (1943) has gone the name of artificial neural networks, reflecting the viewpoint of how learning happens in the brain. Then, Rosenblatt (1958) proposed the concept of the perceptron. Given instances of input from different categories, the perceptron was the first model that learns weights to classify these categories. The backpropagation was then proposed by Werbos (1974a), which realized the multi-layer neural network. In 1985 Rumelhart, Williams, and Hinton introduced backpropagation into the optimization of neural networks, which laid the foundation for the subsequent rapid development of DL. LeCun et al. (1988b) provided the first practical demonstration of backpropagation using convolutional neural networks with the application of reading "handwritten" digits. The most significant breakthrough happened when Hinton et al. (2006) tackled the vanishing gradient phenomena on backpropagation and revealed the potential of DL technology. Now, this technology has begun to rapidly develop owing to 1) the genesis of the big complex and high-dimensional data that cannot be handled well with traditional methods, 2) the consistent development of hardware that allows training very deep networks and growing the size of networks, and 3) the support of the community and big companies that promote the continuous development of this technology. This section gives a gentle introduction to DL techniques and architectures that are used in this chapter.

3.1.1 Learning Algorithms

Machine learning methods generally can be divided into three major categories based on the availability of ground truth data as prior knowledge of the model's output for a given input: *supervised learning*, *unsupervised learning*, and *semi-supervised learning*. In a supervised learning algorithm, the training data are labeled, and the goal is to approximate a function that maps inputs to outputs, given samples of input data and corresponding outputs. The common tasks in supervised learning include classification and regression. In an unsupervised learning algorithm, however, the training data is unlabeled, and the goal is to infer the natural structure existed in

the input data. Clustering, representation learning, and density estimation are the common tasks in unsupervised learning. A semi-supervised learning algorithm falls between these two algorithms, where the goal is to use knowledge learned from a small amount of labeled data to make decisions on a large amount of unlabeled data. This technique requires an assumption to relate the labeled and unlabeled data to justify the conclusions about the unlabeled data from the knowledge learned from a small set of labeled data.

Following in this paragraph, we formally defines supervised learning as the learning algorithm of choice in our analysis. In supervised learning, training data $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ comes in pairs of input features x_i and labels y_i for $i = i, ..., N$, where y typically represents an instance of a fixed set of classes. y can also represent a vector of continuous values for regression purpose. (x_i, y_i) pairs are drawn from unknown distribution $P(x, y)$. The objective is to learn the model parameters Θ through minimizing a loss $\mathcal{L}$ such that for a new instance $(x, y) \sim P$, $f(x; \Theta)$ outputs $\hat{y} \approx y$ with high probability.

3.1.2 Deep Neural Network

A neural network (or artificial neural network) consists of an input layer of neurons (or nodes, units), one or a couple of hidden layers, and an output layer. A hidden layer comprises neurons, weights, and activation units. Weights are defined as $\Theta = \{W, B\}$, where W denotes the set of edges connecting the neurons between consequent layers, i.e. $W = \{w^1, ..., w^l\}$, and B defines the set of biases, i.e. $B = \{b^1, ..., b^l\}$. In a hidden layer, each neuron receives one or more input signals from the raw dataset, or neurons at preceding layers. The linear combination of received signals is passed on to the activation function $\sigma(.)$ to generate the output. The output of the activation function is received by the next layer of the network:

$$f(x) = \sigma(w^T x + b). \tag{3.1}$$

The illustration of a single neuron is shown in Figure 3.1. Multi-layered perceptrons (MLP) are the most well-known traditional neural networks, consisting of several layers of neural units defined in (3.1):

$$f(x; \Theta) = \sigma((w^1)^T \sigma((w^2)^T ... \sigma((w^l)^T x + b^l) + ... + b^2) + b^1), \tag{3.2}$$

where l denotes the layer number. All the neurons in one layer are connected to all the neurons in the preceding and subsequent layers. These types of layers are called fully connected layers. Neural networks containing multiple hidden layers are considered as deep neural networks.

In traditional neural networks, the typical activation function is the sigmoid function that takes a real-valued number x and squashes it into the range between 0 and 1:

$$\text{sigmoid}(x) = \frac{1}{1 + \mathrm{e}^{-x}}, \tag{3.3}$$

Another common activation function is hyperbolic tangent function that takes a real-valued number x and squashes it between -1 and 1:

$$\tanh(x) = \frac{\mathrm{e}^{x} - \mathrm{e}^{-x}}{\mathrm{e}^{x} + \mathrm{e}^{-x}}. \tag{3.4}$$

Sigmoid and hyperbolic tangents place very large negative and positive inputs close to 0 and unity or -1 and 1, respectively, causing saturation in practice. In modern deep networks, the rectified linear unit (ReLU) is commonly used, which is faster to compute the gradient, and less prone to vanishing gradient or saturation.

$$\text{ReLU}(x) = \max\{x, 0\} \tag{3.5}$$

At the last layer of the network, however, the softmax activation unit takes the outputs. The role of the softmax is to normalize the outputs to a probability distribution over the predicted output classes:

$$\text{softmax}(x; \Theta)_c = P(y = c|x; \Theta) = \frac{\mathrm{e}^{w_c^T x + b_c}}{\Sigma_{c=1}^{C} \mathrm{e}^{w_c^T x + b_c}}, \tag{3.6}$$

where w_c presents the vector of incoming weights to the output neuron of class c, and C determines the number of classes. $P(y = c|x, \Theta)$ is called class score, implying the computed probability distribution. It tells how likely input x belongs to the class C. To optimize the parameters, the output probability distribution is compared to the true distribution. This comparison is the role of the loss function that evaluates how well the network classifies the sample x to the corresponding class. The choice of the loss function for neural networks is typically similar to the choice of loss function for other parametric models (as linear models). For instance, cross-entropy, a common

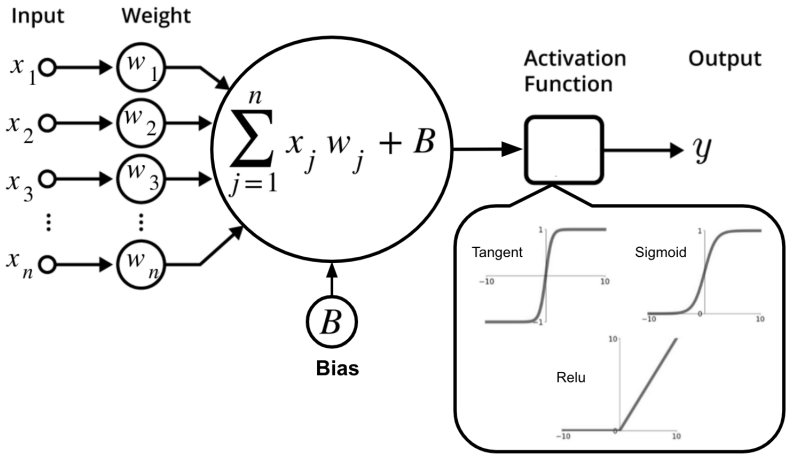


Figure 3.1: Single neuron illustration. Each neuron in a neural network takes one or more input signals x_j that is multiplied by the corresponding weight w_j. The linear combination of received signals is passed on to the activation function to generate the output. Activation functions add nonlinearity to the whole network, which enables the network to learn complex patterns. The output of the neuron becomes the input to the following neurons in the network.

way to measure the distance between two probability distributions, is used as a loss function for classification tasks. The cross-entropy loss is defined in Eq (3.7).

$$\mathcal{L} = -\Sigma_{c=1}^{C} y_c \log P(Y = c|X = x; \Theta) = -\Sigma_{c=1}^{C} y_c \log \frac{e^{w_c^T x + b_c}}{\Sigma_{c=1}^{C} e^{w_c^T x + b_c}} \tag{3.7}$$

The ground truth vector y in DL frameworks is commonly one-hot encoded. It means only the positive class element of the target vector is non-zero. Therefore, the elements of the summation, which are zero due to target labels, can be discarded, and the cross-entropy loss can be rewritten as:

$$\mathcal{L} = -\log \frac{e^{w_c^T x + b_c}}{\Sigma_{c=1}^{C} e^{w_c^T x + b_c}} \tag{3.8}$$

This is equivalent to the negative log-likelihood equation. To find the optimal parameters of the network, we minimize the negative log-likelihood obtained from the entire training data.

$$\mathcal{L}_{total} = -\Sigma_{i=1}^{N} \log(P(y_i|x_i; \Theta)) \tag{3.9}$$

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}}\, \mathcal{L}_{total}(x, y; \Theta), \tag{3.10}$$

where $\mathcal{L}_{total}$ and Θ^* determine the total loss and optimal parameters, respectively. Due to a large number of data points in practice, it is infeasible to learn all parameters at once. Currently, the most popular practical way to fit parameters of a neural network to a dataset is stochastic gradient descent (SGD). SGD iteratively performs an update on every sample, x^i or every mini-batch of n training examples, denoted by $x^{(i:i+n-1)}$:

$$\Theta_t = \Theta_{t-1} - \eta \nabla_\Theta \mathcal{L}(\Theta; x^{(i:i+n-1)}; y^{(i:i+n-1)}), \tag{3.11}$$

where η indicate the learning rate.

Optimization in neural networks is non-convex due to the numerous suboptimal local minima or saddle points. These make the convergence to the global minimum or an optimal local minimum challenging for SGD. Other challenges that need to be dealt with SGD include: speeding up the learning phase, adjusting learning rate η properly, tuning learning rate during training, and adapting the learning rate to the network parameters (that have very different frequencies in the case of a sparse dataset). To address these challenges, many algorithms have been proposed as extensions to SGD, including Momentum (Sutton, 1986), NAG (Nesterov, 1983), Adagrad (Duchi et al., 2011), Adadelta (Zeiler, 2012), RM-Sprop, Adam (Kingma and Ba, 2014), and many others. These methods pick different strategies for updating the weights in each iteration to improve the convergence in a different situation, but the main principle of optimization remains the same. The following section introduces Momentum and Adam, mainly used as DL optimization algorithms in this thesis.

Momentum and Adam

SGD is slow, caused by poor conditioning of the Hessian matrix and variance in the stochastic gradient. The momentum extension is designed to speed up the convergence, for instance, in scenarios where high curvature, small but consistent gradients, or noisy gradients slow down the learning. Momentum circumvents this problem by accumulating an exponentially decaying moving average of past gradients and continues to move in their direction. It simply means momentum memorizes past gradients

to maintain the direction towards the global minimum that decreases oscillations and accelerates learning. To memorize past gradients, a fraction γ of the moving average gradient of the previous time step υ_{t-1} is added to the current gradient:

$$\upsilon_t = \gamma\upsilon_{t-1} + \eta\nabla_\Theta\mathcal{L}(\Theta_t), \tag{3.12}$$

which then is used to update the weights of the network:

$$\Theta_t = \Theta_{t-1} - \upsilon_{t-1}$$

Hyperparameter $\gamma \in [0, 1)$ indicates how quickly the contributions of previous gradients exponentially decay. Larger values of γ relative to η, increase the contribution of the previous gradient to the current direction.

Adaptive moment estimation (Adam) (Kingma and Ba, 2014) is another widely used extension to SGD. This optimizer adapts the learning rate for each neural network parameter using estimations of the first and second moments of the gradient. It performs smaller updates for frequently occurring features, and larger updates for infrequent features. Adam keeps an exponentially decaying average of the past gradients υ_t (similar to momentum) as well as the past squared gradients m_t (similar to RMSprop):

$$\begin{aligned} \upsilon_t &= \beta_1\upsilon_{t-1} + (1-\beta_1)\nabla\mathcal{L}(\Theta_t) \\ m_t &= \beta_2 m_{t-1} + (1-\beta_2)(\nabla\mathcal{L}(\Theta_t))^2, \end{aligned} \tag{3.13}$$

where the hyperparameters $\beta_1, \beta_2 \in [0, 1)$ control the exponentially decaying rates of these moving averages. Adam also considers bias corrections:

$$\begin{aligned} \hat{\upsilon}_t &= \frac{\upsilon_t}{1-(\beta_1)_t} \\ \hat{m}_t &= \frac{m_t}{1-(\beta_2)_t} \end{aligned} \tag{3.14}$$

Since m_t and υ_t are initialized as vectors of zeros, they can be biased towards zero, during the initial time steps and for small decay rates. The weights of the network are updated through the following rule:

$$\Theta_{t+1} = \Theta_t - \frac{\eta}{\sqrt{\hat{m}_t} + \epsilon}\hat{\upsilon}_t. \tag{3.15}$$

In DL models, the best algorithm is the one that can traverse the loss for that problem very well, which is chosen empirically than mathematically. For example, in this thesis, adaptive learning algorithms are shown to be a better choice of optimization for learning mass spectrometry data that is inherently sparse.

Regularization

Overfitting is a common problem of DNNs due to the complexity of the model that tends to fit all data points in the training set completely. Regularization is a technique to alleviate this problem by simplifying the model, which explicitly boost the network's performance on the test phase, possibly at the expense of increased training error. Many forms of regularization are available to the DL practitioner, including ℓ_1 (Lasso) and ℓ_2 (weight decay) parameter norm penalties, data augmentation, adding noise to the weights (Graves, 2011), modeling the noise on the labels (Szegedy et al., 2016), dropout (Srivastava et al., 2014), multi-task learning, early stopping, batch-normalization (Ioffe and Szegedy, 2015a), bagging (Breiman, 1996), and other ensemble methods.

One or a couple of regularization strategies can be applied for training a generalized network simultaneously. For example, early stopping, dropout, and ℓ_2 regularization. Early stopping terminates the weight updates when the minimum validation error is reached. Due to both its effectiveness and simplicity, it is one of the most commonly used forms of regularization in DL. The other most common regularization strategy is the ℓ_2 parameter norm penalty, known as weight decay. This regularization strategy adds a regularization term $\Omega(\theta) = \frac{\lambda}{2}||W||_2^2$ to the objective function. λ is the regularization term, and θ denotes some subset of the parameters, typically targeted weights of the network, not the biases. ℓ_2 drives the weights closer to the origin. ℓ_1 with the same strategy, adds $\Omega(\theta) = \frac{\lambda}{2}||W||_1$ to the objective function, which forces the weight parameters to become zero. Dropout deactivates random weights at every epoch with a certain probability during the training. Each epoch sees a different set of nodes, resulting in a different set of outputs. Therefore, it can be seen as an ensemble technique in machine learning. In general, regularization in neural networks simplifies networks during training to reduce overfitting. For instance, smaller weight parameters in ℓ_1 and ℓ_2 and deactivating neurons in dropout make some neurons neglectable. Consequently, the neural network becomes less complex, less biased to the training data points, and less prone to overfitting.

3.1.3 Convolutional Neural Network

A convolutional neural network or CNN is a special neural network that utilizes the convolutional operation to extract features. The layers of CNN consist of a set of spatial kernels or filters of small sizes which slide over the input data and compute the dot products at each spatial location. These layers are called convolutional layers. The kernels at each layer indicate the weights or parameters of the network that are learned over the training. Compared to fully connected layers, convolutional layers can preserve the spatial structure. Fully connected layers operate on vectors. Therefore, Given an image task, the images are vectorized, which leads to losing the spatial correlation between neighboring pixels. This step is bypassed by convolutional layers using convolution operation on the full structure of the data. At each layer l, a set of K kernels $W^l = \{W_1^l, W_2^l, ..., W_K^l\}$ and a bias b_l convolved with the input of that layer X^{l-1}. The non-linear activation function $\sigma(.)$ takes the results and generates the output feature map X^l:

$$X^l = \sigma(W^l * X^{l-1} + b^l). \tag{3.16}$$

Almost all convolutional networks employ one more stage called the pooling function to further modify the output. A pooling function replaces the values of the feature map with a summary statistic of these values in a rectangular neighborhood. Popular pooling functions include the max-pooling (Zhou and Chellappa, 1988), which takes the maximum values, and average pooling, which takes the mean values of the outputs. The pooling operation leads the network to be approximately invariant to small translations. It also reduces the number of learnable parameters that ease the optimization. The layers on top of the network typically are followed by fully-connected layers. The softmax then takes outputs of the last fully connected layer and generates a probability distribution over classes. The network is then trained using the maximum likelihood principle, similar to the training neural network described in section 3.1.2.

There has been a great effort in using deep neural networks (DNN) since a CNN-based method significantly outperformed other approaches for the first time in the well-known ImageNet challenge (Krizhevsky et al., 2012). Since then, dozens of different network topologies have been proposed to improve the performance of DNNs for various applications, e.g., varying layers and filter sizes (Zeiler and Fergus, 2014;

Simonyan and Zisserman, 2014), development of the inception module (Szegedy et al., 2015) that replaces the mapping defined in Eq (3.16) with a set of convolutions of different sizes, and adding additional connectivity between layers (He et al., 2016a) that ease the flow of the gradient backward through the layers. Furthermore, effects of different training techniques (Hinton et al., 2012b; Huang et al., 2016a), better activation units (Glorot et al., 2011), different stochastic optimization method (Duchi et al., 2011; Kingma and Ba, 2014), faster training methods (Ioffe and Szegedy, 2015b), and different connectivity patterns between layers (Huang et al., 2016b) have improved DNN efficiency.

Parallel to advances in training deep networks, there have been attempts to interpret classification decisions of trained networks and even first steps to go beyond this (Holzinger et al., 2019). We will have a comprehensive review of this topic in the next section.

3.2 Model Interpretation

With the success of machine learning in industry and science, there has been a growing demand for interpreting these models, especially in the medical domain (e.g., healthcare and medicine), involving high stakes decisions that impact human health and life. Gaining a better understanding of ML problem-solving strategies opens better communication so that users may know if and when to trust model predictions. For example, if interpretability explains that the model may make an individual prediction according to relevant variables, the user can more reliably take actions according to the model's prediction. Besides trusting individual predictions, it is essential to measure the robustness of the model before the deployment. To this end, an evaluation is performed using accuracy metrics on the validation dataset. However, real-world data has more variations from validation data; therefore, even with high precision, the evaluation metric on validation may not represent the model generalization. In this case, interpretability is also a worthwhile solution to facilitate debugging, detecting possible biases, and confirming the model generalization.

Model understanding can be achieved by building inherently interpretable models like linear models and shallow decision trees. However, in recent years, the success of more complex models like DNNs encourages scientists to advocate post-hoc explanations through explaining pre-built models. Accordingly, DL libraries have star-

ted to include these methods in their own explainable-AI libraries, such as Pytorch Captum and TensorFlow tf-explain. This thesis uses post-hoc explanation methods to describe DNN decisions. The following subsection defines and introduces various post-hoc explanation methods. It is worth noting that, despite many articles that have attempted to define *interpretability* and *explainability* (Doshi-Velez and Kim, 2017; Ribeiro et al., 2016; Lipton, 2018; Došilović et al., 2018; Arrieta et al., 2020), there is no clear exposition on how they should be incorporated into the great diversity of implementations of ML models. Therefor, this thesis uses the words *interpretability* and *explainability* interchangeably.

3.2.1 What is an Explanation?

The explanation is any interpretable description of decision-maker behavior, which represents the true decision/reasoning process of the model, referred to as *faithfulness*, and is understandable by the user, referred to as *readability*. Being less *faithful* is sometimes a trade-off for more *readability*. However, this can be changed based on the targeted audience and their expertise level. For example, the explanation for text classification can be a binary vector indicating the presence or absence of a word readable by the general audience, even though the classifier may use more complex features such as word embedding. For image classification, likewise, an explanation may be a binary vector indicating the "presence" or "absence" of a contiguous patch of similar pixels readable for general users, while the classifier may represent the image as a tensor with three color channels per pixel. Given this definition, many approaches can fit into a model explanation, e.g., providing model parameters understandable by the users, releasing many example predictions, summarizing the model behavior with rule-based methods or decision tree methods, selecting important features, describing how to flip the model prediction, etc.

The explanation models are grouped into local and global methods. The global explanation describes the model's complete behavior that vets if the model is suitable for deployment at a high level. However, the global explanation might be too complicated to communicate with the user. On the other hand, the local explanation is a more practical explanation approach, which describes the model's behavior in a target neighborhood and measures if individual predictions are made for the right reasons. For instance, in a binary classification setting, the prediction interpretation should highlight discriminating variables between samples of two groups to

show the right reasoning. In medical disease diagnosis, these discriminating variables represent the disease-triggered information. Therefore, the alignment of the model explanation and disease-related information provides evidence of the robustness of the model. Furthermore, in the case of unknown disease-relevant variables, the model explanation can give an insight into the unknown pattern of the data. This thesis elaborates the concepts above by adopting the local explanation methods. The following section introduces different local explanation methods in the literature. To give a broader insight, different methods of global explanation will also be presented.

3.2.2 Local Explanation

Local explanations are designed to describe the individual predictions of the model. The general idea is to approximate a small region of interest in a complex and accurate model to understand why the model arrived at a specific decision. Researchers categorized these approaches differently, based on the research question the explanation is expected to answer: which application does the explanation methods cover? (For example, explaining the image networks through heat-mapping or explaining word embedding networks through feature importance); which one of the predictions, model, or data are being explained; Or based on which concept of the gradient method, perturbation analysis, or variant of backpropagation the methods have been developed? In the following, we categorize the post-hoc local explanation methods based on the concept and intuition of their developments.

Saliency Map

Saliency map is a gradient-based technique proposed by Simonyan et al. (2013a) to generate understandable visualization of deep convolutional network classification models. This method, which is also known as input-gradient, investigates how much a unit change in an input dimension induces in the output. To set up notations, let f map the input data $x \in \mathbb{R}^d$ to the output $y \in \mathbb{R}^C$:

$$f : \mathbb{R}^d \to \mathbb{R}^C. \tag{3.17}$$

Consider this function as a standard supervised classification setting, where C is the number of classes. Input-gradient takes the gradient of class-specific logit - the

output of a model before feeding into the soft-max - with respect to the input i:

$$\nabla_x f_i(x) \rightarrow \mathbb{R}^d, \tag{3.18}$$

which has the same dimension as the input dimension and can be visualized in the form of a heatmap. The highlighted regions on the heatmap demonstrate the most relevant variables or pixels for the model to make decisions. This heatmap in the context of model interpretation is known as a saliency map or sensitivity map.

The input-gradient method, however, is likely to encounter gradient saturation or sensitivity issue. For instance, changing the input from one sample to another, which induces a change in f, does not change the gradient. This problem can arise because the function might be flat around a particular input, or the gradient could be saturated around that point. Besides, the sensitivity maps are often visually noisy, highlighting some pixels that – to a human eye – seem randomly selected. Several methods have been proposed to circumvent the undesired properties of the sensitivity map. One of them is smooth-grad (Smilkov et al., 2017) that enhances the input-gradient method through smoothing the gradient. The idea is to take the image, sample images by adding noise to the input image, and calculate the average of saliency maps of the sampled images:

$$\frac{1}{N}\sum_{i}^{N} \nabla_{(x+\epsilon)} f_i(x+\epsilon), \epsilon \sim \mathcal{N}(0, \sigma^2). \tag{3.19}$$

It is shown that smooth-grad output is visually more coherent than saliency map. Another method that counteracts with sensitivity issue of saliency map is integrated gradient proposed by Sundararajan et al. (2017). Instead of evaluating the partial derivative just at the given input x, integrated gradient computes the average of it while the input is changing along a linear path from a baseline $\tilde{x}$ to the input of interest x:

$$(x - \tilde{x}) \int_{\alpha=0}^{1} \frac{\partial f(\tilde{x} + \alpha \times (x - \tilde{x}))}{\partial x}, \tag{3.20}$$

where $\tilde{x}$ could be a black image for image networks or a zero embedding vector for text models. In practice, this can correspond to interpolating inputs from the baseline, computing the saliency map for all different interpolates, and summing them up.

Modified Backpropagation

Modified backpropagation methods propagate the signal of importance in output neurons backward through all the network layers to the input neurons. It is similar to computing the saliency map using the gradient of output w.r.t input, except for handling the non-linearity at rectified linear units (ReLUs) or the pooling layers. Zeiler and Fergus (2014) proposed a deconvolution network to map all the network activities back to the input, looking for a pattern in the input space. A given activation is propagated back through un-pooling, rectifying, and filtering (transpose of learned features in a forward path) to the input layer. To un-pool for max-pooling in deconvolution network, the switches (the maximum position within each pooling region) are recorded on the forward pass. Besides, the signal comes into ReLU in backpropagation is zeroed out if it is negative:

$$R_i^l = \begin{cases} R_i^{l+1}, & \text{if } R_i^{l+1} > 0 \\ 0, & \text{Otherwise.} \end{cases} \tag{3.21}$$

$l \in [1, L]$ determines the current layer of the network, and R denotes the signal of importance in the backpropagation. By contrast, the input-gradient method zeroed out the signal that comes into ReLU if and only if the incoming signal to the ReLU in the forward pass is negative (normal backpropagation in gradient method):

$$R_i^l = \begin{cases} R_i^{l+1}, & \text{if } f_i^l > 0 \\ 0, & \text{Otherwise} \end{cases} \tag{3.22}$$

where $f_i^{l+1} = \text{ReLU}(f_i^l) = \max(f_i^l, 0)$, $R_i^{l+1} = \frac{\partial f^{out}}{\partial f_i^{l+1}}$. Guided backpropagation (Springenberg et al., 2014) combined these two approaches. It zeroes out the signal of importance coming into the ReLU, if the signal itself is negative in the backward pass or the input to the ReLU is negative in the forward pass:

$$R_i^l = \begin{cases} R_i^{l+1}, & \text{if } f_i^l > 0, \text{ and } R_i^{l+1} > 0 \\ 0, & \text{Otherwise} \end{cases} \tag{3.23}$$

This simple modification in Guided backpropagation obtains a sparser saliency map in comparison to the input gradient and provides a visually better explanation. The limitation of these methods is that negatively contributed samples would be ignored

due to discarding the negative gradient. Besides, discontinuing the gradient might cause undesired artifacts.

Layer-wise Relevance Propagation

Layer-wise relevance propagation (LRP) method (Bach et al., 2015) is the family of explanation methods that uses the layered structure of the neural network to construct the explanation. It distributes the neuron activation of the decision layer to the previous layers until the input layer is reached. The importance signal on the input layer determines how much and to what extent each feature in a particular input contributes to the network's decision. To back propagate the activation, Montavon et al. (2019) specify the variety of rules for controlling the positive and negative relevances. These rules are tuned for explanation quality, e.g., sensitivity in top layers, robustness in lower layers, or tracking the contradicting evidence in the input. The limitation to this method is that the LRP propagation strategy must be adopted to each new architecture, and it makes some assumptions about the structure of the model. (more detail about this method and its variation is delayed to 3.5.3.)

Feature Importance

Despite the LRP method, the feature importance method does not consider any assumptions about the internal structure of the model and can be applied to any machine learning algorithm. Occlusion analysis is one of the approaches in this category that measures the effect of perturbing individual inputs on the later neurons of the network:

$$R_i = f(x) - f(x \odot \text{pert}_i), \tag{3.24}$$

where $pert_i$ is an indicator function to perturb (e.g., to remove), and $\odot$ denotes the element-wise product. The perturbation can be a simple occlusion (Zintgraf et al., 2017; Zeiler and Fergus, 2014), an in-painting occluded pattern using generative models (Agarwal et al., 2019), or a meaningful perturbation that is synthesized (Fong and Vedaldi, 2017). In genomics (Zhou and Troyanskaya, 2015), perturbation is introduced by virtual mutations at individual positions in a genomic sequence. However, such methods can be computationally inefficient or even infeasible for high-dimensional data, since each perturbation requires a separate forward propagation through the network. These methods may also underestimate the importance of

features that have saturated their contribution to the output.

LIME (Ribeiro et al., 2016) is another approach in this category, which is applied to image and text analysis. LIME replaces the decisions of a function with a local surrogate model that is structured so that it can be explained by a self-interpretable model (like the linear model). To this end, LIME first defines some local perturbation $\text{pert}(x_i)$ around a single sample x_i, and then approximates the function f locally around perturbed x_i with the linear model g:

$$\arg\min_{g \in G} L(f, g, \text{pert}(x_i)) + \Omega(g), \tag{3.25}$$

where $\text{pert}(x_i)$ is the local perturbation of input x_i, and $\Omega(g)$ is a regularizer. The coefficient of the linear model g is then served as the explanation. LIME is quite customizable in the different domains since users can specify the perturbation function, the distance/similarity measure, the size of the locality, and the expression of the explanation. Another method in this category is SHAP (Lundberg and Lee, 2017), which provides a tractable approximation to the Shapely value (Shapley, 1953). Intuitively, it works similarly to LIME by perturbing the instances and recording which perturbations lead to a change in the output. Nevertheless, the main difference is that in SHAP, the marginal contribution of input features in all possible perturbations towards the prediction is considered, which is known to be a fair way of attributing predictions to specific features.

It is worth mentioning that the model-agnostic methods like LIME, SHAP, and occlusion models are slower to compute than gradient-based methods and sometimes can be infeasible to calculate in high-throughput data. For instance, in proteomics, sometimes a couple of thousand or millions of features should be analyzed to compute all the importances.

3.2.3 Global Explanation

Global explanations are designed to explain the complete behavior of a given model to provide a bird's-eye view of the model. It can help detect big picture biases persistent across larger subgroups of populations, which are often harder to detect by examining the local explanation of several instances. In this sense, global explanations are complementary to local explanations.

One of the techniques to construct global explanations is the collection of local

explanations, in which a local explanation for every instance in the data is generated. Then, a subset of K local explanations is picked to constitute the global explanation. For example, LIME explains a single prediction for a single instance, but due to the time/patience that human has, it could be impossible to examine all possible explanations to understand the model's global behavior. To address that, SP-LIME (Ribeiro et al., 2016) advocates picking K explanations to show to the users. These K explanations are picked so that they summarize the model's behavior and are not redundant in their descriptions. SP-LIME uses sub-modular optimization and greedily picks k explanation. Besides, it is still model agnostic because it does not require access to the internal details of the underlying predictive model. If we repeat the same procedure but replace LIME with the Anchor algorithm, a global explanation is obtained by presenting a subset of K local rule sets (Ribeiro et al., 2018), which still makes no assumption about the model while providing explanations.

The representation-based approach is another global explanation method that uses an internal representation of a DNN to provide insight into the concepts that the model might have learned. One of the representation-based approaches is network dissection (Bau et al., 2017) that determines the model's reliance on the concept of interest, by quantifying the interpretability of latent representations. To this end, a broad set of human-labeled visual concepts is first identified, and the activations of hidden variables to these known concepts are gathered; then, the alignment - intersection over union- of the activation and ground truth label are quantified. This method can explain globally how much the model relies on each concept of interest. For instance, how much a model relies on the sky scene to classify tall buildings. While this method encodes hidden units for a single concept, compositional explanation (Mu and Andreas, 2020) encodes hidden units for the composition of concepts. For instance, the sky scene, spike shape, and water tower are encoded for classifying tall buildings.

TCAV (Kim et al., 2018a) is another method that uses the model internal representation, but for measuring the sensitivity of a model's prediction to user-provided concepts. For instance, it tells how sensitive a prediction of "zebra" is to the presence of strip. TCAV works by first collecting the images from the concept of interest and a different set of random images. Second, the images are fed into the network to collect activations across different layers, which then are used for training a linear model to separate the concept of interest. Finally, the trained linear model encodes for the concepts, in which the weights are used for measuring the sensitivity of the

model prediction. This is computed using directional derivatives in the direction of the concept of interest. While TCAV seeks a discriminator between two concepts, RCV (Graziani et al., 2018) seeks the direction of the greatest increase in the measures for a single continuous concept, which extends this method to the regression task. This method is applied to identifying the factors relevant to the classification of breast cancer histopathology.

Table 3.1 lists the explanation methods based on interpretability methods used or interpretability mechanisms.

Table 3.1: Summary of explanation methods arranged according to interpretability methods used, or interpretability mechanism. Local (L) means if the methods are the local explanation that describes individual predictions, and global (G) means if the methods are the global explanation that explains the complete behavior of a given model. Ad-hoc (AH) refers to the models that are inherently interpretable, and post-hoc (PH) describes methods that explain pre-built complex models. Model-agnostic ✓means if the interpretation is independent of the underlying learning model.

Explanation Method	References	Local (L)/Global (G)	Ad-hoc/ Post-hoc	Model-Agnostic	Explanation Mechanism
Decision trees	García et al., 2009 Hara and Hayashi, 2016	G	AH	-	Inherentely
Linear models	Haufe et al., 2014	G	AH	-	
Input gradient	Simonyan et al., 2013a	L	PH	✗	Sensitivity analysis
Smooth-grad	Smilkov et al., 2017	L	PH	✗	
Deconvolution	Zeiler and Fergus, 2014	L	PH	✗	Backpropagation
Guided backpropagation	Springenberg et al., 2014	L	PH	✗	
LRP (Layer-wise relevance propagation)	Bach et al., 2015 Montavon et al., 2019	L	PH	✗	
Occlusion	Agarwal et al., 2019 Fong and Vedaldi, 2017 Zintgraf et al., 2017	L	PH	✓	Perturbation analysis
LIME/SP-LIME	Ribeiro et al., 2016	L/G	PH	✓	
SHAP (Shapley additive explanations)	Lundberg and Lee, 2017	L	PH	✓	
Anchore	Ribeiro et al., 2018	G	PH	✓	
Network dissection	Bau et al., 2017 Mu and Andreas, 2020	G	PH	✗	Representation analysis
TCAV	Kim et al., 2018a	G	PH	✗	
RCV	Graziani et al., 2018	G	PH	✗	

3.2.4 Medical Setting Application using Explanation Methods

The explanation methods appear for different purposes, e.g., to give a justification and reliance to a model's decision-making, to improve the system by providing the reasoning for the human or machine on an ongoing iteration, to control a system by giving an insight to the bugs in low critical scenarios and make it more robust, and to discover and gain knowledge from a system. Although these applications capture different motivations for explainability, they share similar concepts, and one application may include more than one purpose. In the following, we review some of the recent applications in medical settings.

Guided backpropagation (Springenberg et al., 2014) was employed by Larson et al. (2018) to *reason* the assessment of skeletal maturity on pediatric hand radiographs with rivaling performance that of expert radiologists.

Rieger et al. (2020) proposed a skin cancer detection model based on contextual decomposition explanation penalization (for *reasoning and improvement*). This method uses contextual decomposition of the feature importance to explain the DL model. During training, the explanation penalizes data points and their prediction labels in loss function to align the predictions with prior knowledge. This approach improves the skin cancer diagnosis model that is less reliant on spurious correlation.

Sayres et al. (2019) employed an integrated gradient heat map as an assistant tool (*discovery purpose and improvement*) for diabetic retinopathy diagnosis. They examined the interaction of physicians with different DL prediction explanations. They suggested that by increasing the transparency, a model assistant, can boost ophthalmologist performance beyond what can be achieved by model only or ophthalmologist alone. In another study of diabetic retinopathy, TCAV explanation (Kim et al., 2018a) was employed to assess the model reliance on clinically relevant factors. They discussed that TCAV might be useful for helping experts interpret and fix model errors when they disagree with model predictions (*reasoning and controlling* purposes).

Thomas et al. (2019) utilized the LRP explanation method to explain the prediction of cognitive states from fMRI data in order to identify the physiologically appropriate brain regions associated with these states (*discovery purpose*). LRP analysis was performed on the level of a single input sample, enabling an analysis of the fine-grained temporospatial distribution of brain activity underlying sequences

of single fMRI samples.

Several challenges emerge with the application of explanation methods, especially in medical settings. Several explanation methods have been developed and tailored towards standard architectures -Vgg16, ResNet50, and Inception - on standard datasets like Imagenet. However, adapting these methods to high-throughput data, e.g., proteomics, is challenging due to the homogeneity of the inputs -where the focuses are more on similar artifacts or same poses, but the conditions are quite different from patient to patient- which will be discussed in this chapter.

3.3 Transfer Learning

One of the promising practical concepts in DL is transfer learning. The idea is to take the neural network's knowledge from one task and apply that knowledge to a separate task. For example, a neural network that has learned to recognize objects (e.g., cats, table, glass) in images, uses that knowledge or part of that knowledge to read and analyze X-ray scans. Transfer learning, which focuses on transferring the knowledge across domains, might be inspired by the human ability to utilize previous experience to learn a new task faster with less effort. For example, a person who has learned inline skating can learn ice skating faster than others since both may share some common knowledge, such as a sense of balance on skates.

The need for transfer learning has been brought by the ML and DL hunger for abundant labeled training instances having the same distribution as the test data. This need is more pronounced in medical applications since collecting sufficient training data in this domain can be more expensive, time-consuming, or even infeasible. Approaches like semi-supervised learning (Chapelle et al., 2009) address this problem by increasing the learning accuracy using unlabeled data. In semi-supervised learning, both the labeled and unlabeled instances are drawn from the same distribution. In opposition to semi-supervised learning, in transfer learning, the data distributions of the source and the target domains can be different. Transfer learning tries to leverage the knowledge of a different but related domain to compensate for the insufficient amount of labeled data. In many medical cases, transfer learning is preferred because collecting unlabeled data is also strenuous and unrealistic.

One closely related area to transfer learning is multi-task learning (Ruder, 2017), where the idea is to solve several related tasks simultaneously. Multi-task learn-

ing shares the representations between the tasks and leverages the domain-specific information contained in the training signals of these tasks to obtain better generalization. While both transfer learning and multi-task learning improve performance by transferring knowledge, transfer learning optimizes a single loss, but multi-task learning optimizes more than one loss. Although this thesis only focuses on transfer learning, Our proposed method can be extended to multiple parent tasks and formalized as multi-task learning.

Transfer learning can improve the performance of DL models in three ways: (1) improve the performance in the initial epochs of the training model, (2) accelerate the training phase, and (3) improve the overall performance.

To formally define the transfer learning, some notations need to be introduced. Let $\mathcal{D} = \{\chi, P(X)\}$ represent a domain, which contains the feature space χ, and marginal probability distribution $P(X)$, where $X = \{x_1, x_2, ..., x_n\} \in \chi$, and $\mathcal{T} = \{y, f(x)\}$ represent a task, which contains label space y and target prediction function $f(x)$. Some machine learning models output the predicted conditional probability of instances, $f(x) = P(y|x)$. Transfer learning then can be defined as follows:

Definition 3.3.1 (Transfer Learning)**.** *Given observations in a task $\mathcal{T}_s$ based on a source domain $\mathcal{D}_s$, and observations in a task $\mathcal{T}_t$ based on a target domain $\mathcal{D}_t$, transfer learning utilizes the latent knowledge from $\mathcal{D}_s$ and $\mathcal{T}_s$ to improve the performance of predictive function $f_{\mathcal{T}}(.)$ for learning $\mathcal{T}_t$, where $\mathcal{D}_s \neq \mathcal{D}_t$ and/or $\mathcal{T}_s \neq \mathcal{T}_t$. Typically, the number of observations in $\mathcal{D}_s$, N_s, is much larger than the number of observations in $\mathcal{D}_t$, N_t; $N_s >> N_t$.*

In this thesis, deep transfer learning is used to address the limitation of labeled training data, which is defined as follows:

Definition 3.3.2 (Deep Transfer Learning)**.** *Given $\mathcal{D}_s$, $\mathcal{T}_s$, $\mathcal{D}_t$, and $\mathcal{T}_t$, a task is defined as deep transfer learning if $f_{\mathcal{T}}(.)$ is a non-linear function and reflects a deep neural network.*

Tan et al. (2018) divided deep transfer learning into four widely accepted categories: instance-based deep transfer learning, mapping-based, deep transfer learning, network-based deep transfer learning, and adversarial-based deep transfer learning. We first introduce these categories, and then review the applications in network-based deep transfer learning, which is mainly used in this thesis.

3.3.1 Instance-based Deep Transfer Learning

In instance-based deep transfer learning, partial instances from the source domain are used to supplement the training set in the target domain. Appropriate weight values are assigned to these selected instances using a specific weight adjustment technique. The partial instances from a domain with the appropriate weight added to the target training set boost the model performance.

3.3.2 Mapping-based Deep Transfer Learning

In mapping-based deep transfer learning, the instances in the source domain and target domain are mapped into new data space, in which the instances of these different domains are more similar. In the new space, the instances from both domains are used for a union deep neural network.

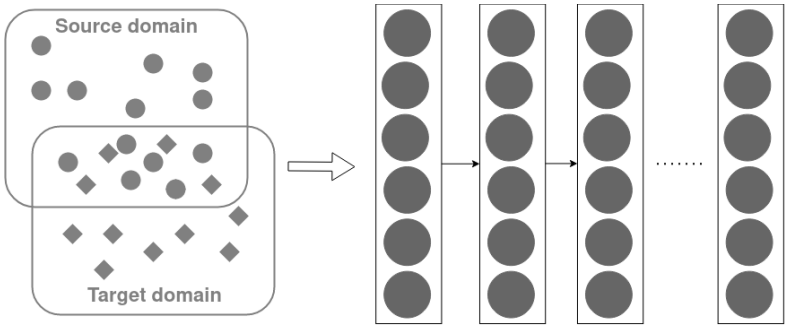
3.3.3 Network-based Deep Transfer Learning

In network-based deep transfer learning, the partial network pre-trained in the source domain is used to retrain the target domain. Similar to human cognition, neural networks tend to learn fine-grained to coarse-grained features, progressively through the early layers to top layers. The fine-grained features that are typically similar across the source and target domains are considered as the transferable knowledge. Transferring the knowledge in this way facilitates the learning process of the target task, since a part of the network has already trained on the source data. Therefore, the network may faster or easier reach the maximum performance.

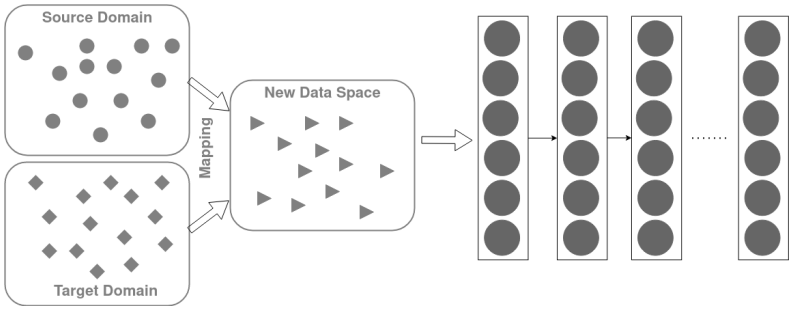
3.3.4 Adversarial-based Deep Transfer Learning

In adversarial-based deep transfer learning – inspired by generative adversarial nets – a transferable representation is found such that it is discriminative for the main learning task and indiscriminate between the source domain and target domain. The front layers of a network are considered feature extractors. The features are sent to the adversarial layer, which aims to discriminate the origin of the features, in terms of whether they are extracted from the domain source or the target source. The worse performance of adversarial layer indicates a small difference between the two types of feature and better transferability, and vice versa. In the course of training,

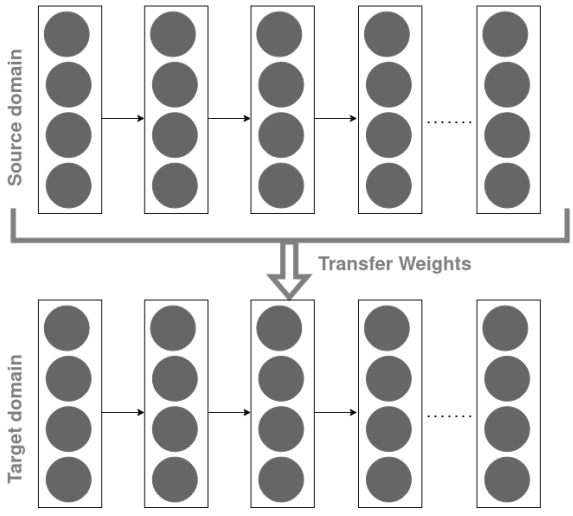
the transfer network is forced to learn general features with more transferability. The sketch map of these four techniques is shown in Figure 3.2.



(a) Instance-based deep transfer learning.

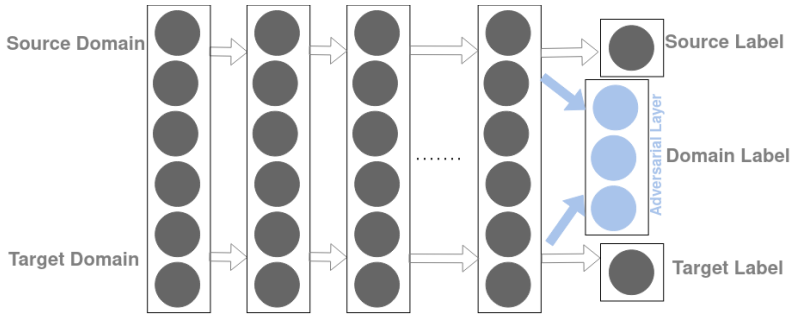


(b) Mapping-based deep transfer learning.



(c) Network-based deep transfer learning.

See the next page for a detailed description.



(d) Adversarial-based deep transfer learning.

Figure 3.2: The sketch map of deep transfer learning techniques. These methods are introduced in Section 3.3. In this thesis, we mainly employ network-based deep transfer learning.

3.3.5 Medical Setting Application using Transfer Learning

In the medical domain, for using network-based deep transfer learning, a pre-trained network on a source domain (e.g., the ImageNet dataset consists of fourteen million annotated images with more than twenty thousand categories (Russakovsky et al., 2015)) is fine-tuned on instances of the target domain. One way for fine-tuning is to run the whole pre-trained network on the instances of the target domain for a couple of epochs. Alternatively, the early layers of the pre-trained network are frizzed, and the top layers are retrained on the target instances. Choosing the fine-tuning approach depends on how close the distribution of the source domain instances is to the distribution of the target domain instances. Maqsood et al. (2019) employed transfer learning for multi-class classification of Alzheimer's disease. They used AlexNet (Iandola et al., 2016) whose convolutional layers were pre-trained on the Imagenet dataset, but the top layers, including fully connected and softmax layers, were initialized randomly. The whole modified network was then fine-tuned on Alzheimer dataset. Their experimental results showed the highest accuracy for Alzheimer's stage detection. Similar to this work, Shin et al. (2016) fine-tuned a pre-trained DL model to address the computer-aided detection problems. Transfer learning was adapted to active learning by Tang et al. (2018) to address the classification of various medical data. The idea in this study was to iteratively query a small number of informative unlabeled target samples, remove the source samples which do not fit with the posterior probability distributions in the target domain, and combine the

basic idea of transfer learning with active learning.

One well-known task in bioinformatics is gene expression analysis that was addressed by employing transfer learning to associate gene expression to phenotype (Petegrosso et al., 2017). One of the main challenges in this application is the data sparsity since often, the number of known associations is little. Petegrosso et al. (2017) showed that transfer learning can bring useful training information across human phenotype ontology and gene ontology, and also how multi-task learning could result in an overall improvement by combining relevant training associations and the predictions along with the ontology structure. Xu et al. (2010) applied transfer learning to solve protein-protein interaction prediction task by transferring the linkage knowledge from the source protein-protein interaction network to the target one.

3.4 Proteomics Data Analysis and Related Works

Now that we have introduced all the prerequisites, we can introduce how we address high-throughput data analysis using DL, transfer learning, and the principle of interpretation strategy in proteomics. This section first introduces proteomics data analysis, its challenges, and prior works. Then, it proposes a new method to classify and extract biomarkers from proteomics data using DL methods to circumvent the difficulties associated with the proteomics data analysis.

High-throughput omics methods such as proteomics are often used in various settings to gain a better understanding of the molecular background of human diseases. Due to the precise and fast quantification process, it is widely used in high-throughput proteomics applications (Wang et al., 2016; Hoffmann et al., 2019; Souza et al., 2017). Various levels of data acquisition workflow and the large range of expressed protein abundances in mass-spectrometry make the data highly complex and contain a high level of noise, which are typically handled with different filtering steps and statistical analysis. This requires various manual parameter tuning by experts, which can be different across different labs, times, and interpreters. Automation of this processing can lead the proteomics analysis towards more reproducibility and full automation in clinical decision support systems (CDSS) (Marrugal et al., 2016; Aebersold and Mann, 2003). In this chapter, we aim to develop two demanding and important disciplines for facilitating diagnosis and prognosis in CDSS: 1) the

classification of clinical states from the big data generated by high-throughput data, and 2) the identification of so-called biomarkers.

Classification: The aim of classification in our task is to determine the medical state of a protein mixture sample. What makes this task challenging for ML researchers is the high-dimensionality of data, often outnumbering the sample size. It causes the curse of dimensionality and overfitting in the ML model. The reason is that as the number of dimensions or features in the data grows, the sparsity of the data increases, and the amount of data that is needed to accurately generalize the model grows exponentially. The key to fitting a machine learning model on raw data is to have enough data for the learning process so that they fill the space where the model must be valid. In practice, however, it can be difficult to acquire this amount of real clinical data for a specific disease and especially for mass-spectrometry samples. Please see Section 2.1.2 for an introduction to the mass-spectrometry data acquisition and specifications.

To avoid overfitting, machine learning models can be equipped with preprocessing steps on the data, such as dimension reduction (Meng et al., 2016) or feature selection (Chandrashekar and Sahin, 2014; Espadoto et al., 2019) approaches. Dimension reduction methods, e.g., principal component analysis (Wold et al., 1987) as one of the traditional methods, convert the data into lower dimensional variables by transforming the data into the most informative space. This allows the use of fewer dimensions, which are almost as informative as the original data.

Dimensions or features on the mass-spectrometry data present the ion counts and their masses in the data, which appear as peaks on a plotted spectrum. By that definition, dimension reduction can be seen as peak picking. As peak picking is performed prior to the decision-making analysis, the dimensions or peaks that are left out in the preprocessing step are not further analyzed. This can raise the risk of losing relevant features with low intensities, which are not captured by means of dimension reduction methods. Feature selection approaches (Dash and Liu, 1997; Chandrashekar and Sahin, 2014; Espadoto et al., 2019) pose similar pitfalls, as they are either not powerful enough to not raise the risk of losing relevant biological information, or not computationally efficient, especially when they are coupled with modern ML methods. Many state-of-the-art model-based methods such as SVM (Cortes and Vapnik, 1995), Lasso (Friedman et al., 2010), or ElasticNet (Zou and Hastie, 2005) have been adapted to classify and select discriminating features from

raw MS data (Liu et al., 2009). Other approaches include SPA (Conrad et al., 2017) addressed classification and feature selection using compressed sensing (Donoho et al., 2006) or rule mining approaches (e.g., (Jayrannejad and Conrad, 2017)) where relevant features are identified by adapting a disjunctive association rule mining algorithm to distinguish emerging patterns from MS data. However, analyzing high-dimensional data in their raw format brings the need for scalable models in data handling and model training levels. Towards this end, the research to date has tended to integrate the advantage of DL scalability to different biomedical areas. So far, however, little attention has been paid to using DL to study raw MS proteomics data – mainly due to the lack of enough samples to train a deep network and the lack of enough evidence to guarantee the robustness. We will demonstrate how we robustly train a classifier on two high-throughput modalities of MALDI-MS and LC-MS datasets by leveraging the transferring representation or transfer learning in the context of proteomics data analysis. Despite the prior works, we use synthetically generated data as the source data, produced in an arbitrary number of samples.

Biomarker Detection: The aim of biomarker detection – also known as feature selection – in our context is to discover the identification of proteins that can determine a specific medical condition. Biomarkers in this study are differentially abundant single peaks specified by m/z in MALDI-TOF MS data and pairs of $m/z, RT$ on raw LC-MS map. An advantage of biomarker detection is that a medical condition can be determined by focusing on the biomarker-related areas, reducing the computational cost, and conserving time. Conventional biomarker discovery tools (Bellew et al., 2006; Pluskal et al., 2010; Smith et al., 2006; Qi et al., 2012) often start with a peak detection step to extract interesting and informative areas due to the difficulties of processing noisy, sparse, and high-throughput raw samples. Some well-known software for peak detection are the MsInspect (Bellew et al., 2006) which identifies peaks using a wavelet additive decomposition, the MZmine 2 (Pluskal et al., 2010) which applies a deconvolution algorithm on each chromatogram to detect peaks, and the Progenesis LC-MS (Qi et al., 2012) that uses a wavelet-based approach in such a way that all relevant quantitation and positional information are retained. Other frameworks include XCMS (Smith et al., 2006) in which the peak detection step is addressed by developing a pattern matching approach on overlaid extracted ion chromatograms with Gaussian kernels; AB3D (Aoshima et al., 2014) which iteratively takes the highest intensity peak candidates and heuristically keeps or removes

neighboring peaks to form peptide features; MSight (Palagi et al., 2005) which adapts an image-based peak detection on the generated images from LC-MS maps; and MaxQuant (Cox and Mann, 2008) in which a correlation analysis involving a fit to a Gaussian peak shape is applied. Subsequently, the detected peaks are used for biomarker detection through a combination of several steps, including noise reduction, deisotoping, deconvolution, RT alignment (Listgarten et al., 2007; Podwojski et al., 2009; Gupta et al., 2019), data normalization (Välikangas et al., 2018), data filtering (Schiffman et al., 2019), baseline correction, and peak grouping. However, it is likely to miss low-intensity peaks through different levels of processing. Moreover, the tuned parameters may need to be adjusted again for any data from new sources. This chapter presents a biomarker detection approach that reaches overall better performance than mentioned conventional biomarker approaches independent of the aforementioned preprocessing steps.

The success of DL-based methods, often replacing state-of-the-art classical model-based methods, in many fields such as medical imaging (Lundervold and Lundervold, 2019), biomedicine (Mamoshina et al., 2016), and healthcare (Miotto et al., 2018), has also encouraged the use of DL models for proteomics data analysis. To name a few, DeepIso (Zohora et al., 2019) that combines a convolutional neural network (CNN) with a recurrent neural network (RNN) to detect peptide features; DeepNovo (Tran et al., 2017) and DeepNovo-DIA (Tran et al., 2019), which use DL-based approach (CNN coupled with RNN) for peptide sequencing on data-dependent acquisition (tandem mass spectra) and data-independent acquisition MS data, respectively; pDeep (Zhou et al., 2017b) that adapt the bidirectional long short-term memory for the spectrum prediction of peptides; and DeepRT (Ma et al., 2018a) that employs a capsule network to predict RT by learning features of embedded amino acids in peptides.

Despite the current successful DL approaches on analyzing LC-MS proteomics, most of the studies are empirically driven, and having a justifiable interpretation foundation is largely missing (Iravani and Conrad, 2019). Addressing this issue and adopting DNN interpretability is the focus of this chapter. DNN interpretation was introduced in detail in Section 3.2. Thereby, we explain how explainability provides information about what makes a network arrive at a certain decision. We also reviewed the state-of-the-art explanation method and categorized ad-hoc and post-hoc explanations based on how their algorithm has been built. As in this chapter we utilize post-hoc explanations to investigate the underlying of the data, we once again

categorize this family of explanations based on what we can learn from them about the data: (1) the *function* analysis explains the DL model itself through gradient and shows how much changes in input pixel affect the output (Simonyan et al., 2013b; Smilkov et al., 2017), (2) the *attribution* method interprets the output of the model and explains which input features and to what extent they contribute to the model's output (Sundararajan et al., 2017; Shrikumar et al., 2016; Bach et al., 2015; Sundararajan et al., 2017), (3) the *signal* method tries to find patterns in inputs on which the decision is based (Zeiler and Fergus, 2014; Springenberg et al., 2014; Kindermans et al., 2017), and (4) the *perturbation* analysis calculates the importance of features through measuring the effect of perturbing the elements of inputs on the output (Zintgraf et al., 2017; Agarwal et al., 2019; Fong and Vedaldi, 2017). Although the application of DNN explanation employing perturbation analysis has previously been studied in metabolomics, (Date and Kikuchi, 2018) this explanation is computationally infeasible for high-throughput MS analysis. Hence, the perturbation explanation method is excluded in our high-throughput proteomics analysis.

This section proposes a biomarker detection approach based on interpretable DL to allow analyzing and – ultimately – understanding LC-MS data. The basic idea is as follows: Given two groups of LC-MS samples (say, healthy and diseased), a DNN is trained, and the learned parameters are interpreted through the layer-wise relevance propagation (LRP) technique. We use the result from the interpretation step to identify the areas in the input data that play a crucial role in differentiating the two groups. Our approach is further analyzed to firstly verify the robustness of the network and secondly to detect the differentially abundant peaks as biomarkers. Our biomarker detection model benefits from optimizing class labels rather than expensive annotations at peak levels. We evaluate the proposed model also on real-world data and demonstrate its superiority compared to conventional biomarker detection frameworks. One of the major advantages here is that our method does not depend on the otherwise necessary preprocessing steps. Nevertheless, the preprocessing approaches (Liu et al., 2020a; Kantz et al., 2019) could be potentially added to our framework for further improvement.

Our contribution in this section lies in the combination of the following triad:

(1) Develop a new DL-based classification and feature selection model on high-dimensional raw MS proteomics data.

(2) Tackle the small sample size of real clinical data by integrating the transfer

learning and leveraging synthetically generated data.

(3) Adapt DL interpretation methods to provide explanation and transparency to the model and realize biological relevant information from the data.

3.5 New Feature Selection Method for Proteomics Data

3.5.1 Problem Formulation

Let $\mathcal{I}_n \in \mathbb{R}^D$ for $n = \{1, .., N\}$, and $\mathcal{O}_n \in \{0, 1\}$ be the classifier input vectors in a very large D-dimensional feature space and the corresponding class labels, respectively. Each dimension of the data represent ion-counts demonstrating the features. The aim is to find a small (if possible, minimal) sized subset of features from the input data $\hat{\mathcal{I}} \in \mathbb{R}^d$ ($d << D$), which can be used to build a classifier f. Ideally, f - which is based only on a subset of all available features – possesses the same classification performance as a classifier based on all features. Our approach for feature selection makes use of interpretability analysis for DNNs. Our strategy is to design a DNN architecture, modeled as function f, classify samples into two classes, and learn from the prediction behavior to detect the most d discriminating features. Mathematically speaking, a DNN with L layers can be abstracted as $f(I) = f_L \circ ... \circ f_1(I)$ where each layer is a linear function followed by a nonlinear activation function, such as the rectified linear unit (ReLU (Nair and Hinton, 2010)). Please see Section 3.1 for a gentle introduction to deep neural networks. The power of DNN prediction comes from combining many layers, which at the same time makes it complex and consequently difficult to interpret. The last layer of the trained network contains the class probabilities of the given input data. This information is propagated back through the network to the first layer using LRP. We use this information to identify the parts of a given input that contribute the most to the DNN classification decision over all the training data and determine the discriminating features candidate.

3.5.2 DNN for Proteomics Data Classification

DNNs are characterized by the depth and width of the layers. Depth refers to the number of layers, and width determines the number of neurons on those layers. Depth

and width are selected depending on the complexity of the task, while more neurons usually lead the network to learn more complex functions. Our experiments with DNNs of different depth and width show that even though mass spectrum samples can be classified with only a few DNN layers, using more layers leads to a decreasing generalization error. However, we observe that almost all architecture, ranging from shallow to deep networks, fails to generalize correctly due to the limited available labeled spectra in public datasets. To circumvent this challenge, we integrate the idea of network-based transfer learning to improve the network's performance. The idea, as it is described in Section 3.3, is to take the representation of a neural network that has learned from one task and transfer that representation to the target task. Here, we learn the representation of mass-spectrometry data by first training the network on a large number of instances that we synthetically generated. We generate the synthetic data with specifications similar to the real data, including the baseline noise, mean, and variance of peaks. We use the Maldiquant library (Gibb and Strimmer, 2012) in R to simulate the needed labeled data. The simulated data contains two classes representing diseased and healthy instances, and a network with multiple fully connected layers, followed by ReLU, which adds non-linearity and, consequently, more complexity to the network.

In addition to the proper architecture, training the DNN demands setting up some hyperparameters that – along with the selected architecture – lead to convergence. The hyperparameters include the learning rate l_r, optimization method of gradient descent, and fitting batch size. Setting up the appropriate depth, width, activation function, and hyperparameters leads to high classification performance on the simulated dataset. Consequently, the weight of the trained network or the representation of the synthetic data is then used to initialize the weight of the network for the target task. We then retrain the whole network on the mass proteomics data, which results in a robust and generalized network.

3.5.3 DNN Interpretability for Feature Selection

In most MS proteomics datasets, the number of samples is a lot smaller than the number of features ($N << D$). Most of these features in high-dimensional and sparse proteomics data make no contribution to the decision-making and are redundant. Our strategy to identify the most informative features, which leads to detecting the biomarker candidates, is to learn from the decisions by the machine through the

means of interpretability. A proper interpretation of a classification network gives the user the information about why samples in a certain class are discriminated from other classes. This information, which is associated with the input data, determines features that are more important for the classifier to make decisions. We investigate this information using post-hoc explanation methods, which interpret a single sample. To obtain features that are discriminating for the whole data in general, we run the explanation analysis on the entire dataset on which we base our feature selection method. Note that with this analysis, we can also obtain a bird's eye view of the model behavior to check the robustness of the classifier.

Layer Wise Relevance Propagation

Here, we elucidate our model based on LRP post-hoc explanation. Note that we also set up other explanation methods in the model whose performances are compared in Section 3.6.

LRP (Bach et al., 2015) is a technique to explain a classifier through identifying the contribution of features in an input space in making classification prediction. Given the trained network f and the single sample I, the aim of LRP is to assign each dimension d of I as a relevance score R_d such that

$$f(x) = \sum_{d=1}^{D} R_d^1, \tag{3.26}$$

where R_d should follow qualitative interpretation, i.e., $R_d > 0$ denotes the positive contributions of the presence of dimension d for classification decision, and $R_d < 0$ the negative contributions. LRP leverages the layer-wise structure of the network to compute relevance values. It propagates back the last layer relevance, which is the classifier output $f(x)$, layer by the layer into the input layer, consisting of all the features, to yield R_d^l for $d \in [1, D]$. The class score is maintained through the hidden layers as follows:

$$f(x) = ... = \sum_{d \in l} R_d^l = \sum_{d \in l+1} R_d^{l+1} = ... = \sum_{d}^{D} R_d^1, \tag{3.27}$$

Iterating Eq (3.27) from the last layer down to the input layer I then yields the desired Eq (3.26). To guarantee a unique and meaningful interpretation of the classifier prediction, Bach et al. (2015) define a further constrain to Eq (3.26) and (3.27). It

is assumed that we know the relevance $R_j^{(l+1)}$ of a neuron j at network layer $l+1$ for the classification decision f(x). To start the decomposition, the part of the output corresponding to the targeted class is considered as the relevance value of the last layer. This relevance is decomposed into messages $R_{i\leftarrow j}^{(l,l+1)}$ sent to those neurons i at the layer l which provide inputs to the neuron j. The relevance of any neuron i at the layer l $R_i^{(l)}$ – except the last layer – is defined as the sum of all incoming messages $R_{i\leftarrow j}^{(l,l+1)}$ from neurons j at the layer $l+1$:

$$R_i^{(l)} = \sum_{\text{i: i is input for neuron j}} R_{i\leftarrow j}^{(l,l+1)}. \tag{3.28}$$

Using this definition, the Eq (3.29) is a sufficient condition to ensure maintaining the Eq (3.27).

$$R_j^{(l+1)} = \sum_{\text{i: i is input for neuron j}} R_{i\leftarrow j}^{(l,l+1)} \tag{3.29}$$

Eq (3.28) and (3.29) define the decomposition of the relevances from layer $l+1$ to layer l.

The variant of LRP differs in decomposition rule, which is the way of computing the messages $R_{i\leftarrow j}^{(l,l+1)}$. One possible choice of relevance decomposition is based on the ratio of local and global pre-activations that is given by (3.30), called $LRP.\epsilon$ rule.

$$R_{i\leftarrow j}^{(l,l+1)} = \begin{cases} \frac{z_{ij}}{z_j+\varepsilon}.R_j^{(l+1)}, & \text{if } z_j \geq 0 \\ \frac{z_{ij}}{z_j-\varepsilon}.R_j^{(l+1)}, & \text{otherwise} \end{cases} \tag{3.30}$$

where $z_{ij} = O_i w_{ij}$, $z_j = \sum_i z_{ij} + b_j$, $O_j = g(z_j)$ is the output of the activation function, and w_{ij} defines the weight that connects the neuron j in layer $l+1$ to the neuron i in layer l. The variable ϵ in the denominator is a "stabilizer" term to avoid numerical degeneration when z_j is close to zero. For each layer, R_i is calculated for $i = 1, ..., \text{num\_neurons}$, where num_neurons denotes the number of neurons. The propagation procedure terminates once the input layer has been reached.

Alternatively, the LRP.$\alpha\beta$ rule according to Eq (3.31) allows controlling the importance of positive and negative values that leads to demonstrate contradicting evidence in the input (such that $\alpha - \beta = 1$) . They are typically chosen as $\alpha = 2$ and $\beta = 1$.

$$R_{i\leftarrow j}^{(l,l+1)} = R_j^{(l+1)}.(\alpha.\frac{z_{ij}^+}{z_j^+} + \beta.\frac{z_{ij}^-}{z_j^-}), \tag{3.31}$$

where "+", "−" denote the positive and negative parts. For $\alpha = 1$, $\beta = 0$ the propagation rule is equivalent to LRP.z^+ rule as in Eq. (3.32).

$$R_{i\leftarrow j}^{(l,l+1)} = R_j^{(l+1)} \frac{z_{ij}^+}{z_j^+}. \tag{3.32}$$

Iterating every equation down to the first layer yields the relevance scores of all input dimensions R_d^1.

Feature Selection

R_d^1 gives a score for each dimension of the input vector, demonstrating their strength in decision making. It means that the values assigned to each dimension indicate the importance of these features on the overall classification decision. Therefore, the high-ranked dimensions represent the most discriminating features. Considering offsets, the presence of noise, and different peak indices on samples belonging to different categories, we look through the entire sample relevance distributions, R_{dn}^1 for $n = 1, ..., N$. The normalized relevance values are added up through the entire dataset. The high weighted dimensions show the strength of each individual feature to differentiate the classes. However, for MS proteomics data, in most cases, the identified features are wide, and all the sounding indices are assigned with high values as well (see Fig. 3.4). We establish a postprocessing step to detect the strongest individual features locally to deal with this effect. The postprocessing works as follows: we first select the best feature in the whole spectra, which is determined by weights from the relevance values. Then, the neighbor's features in the determined window are removed. We then select the second-best feature and iterate the process until a stopping criterion is met, e.g., when the classification reaches the whole data classification accuracy.

3.6 Results on MALDI-MS Data

In this section, the aim is to evaluate the proposed method for analyzing high-throughput proteomics datasets. One of the datasets we evaluated our method on consists of some known peptide biomarkers injected into the samples prior to the mass analysis. These biomarkers appear as peaks on the mass spectrum whose masses are predictable. We discover the strength of our feature selection or biomarker

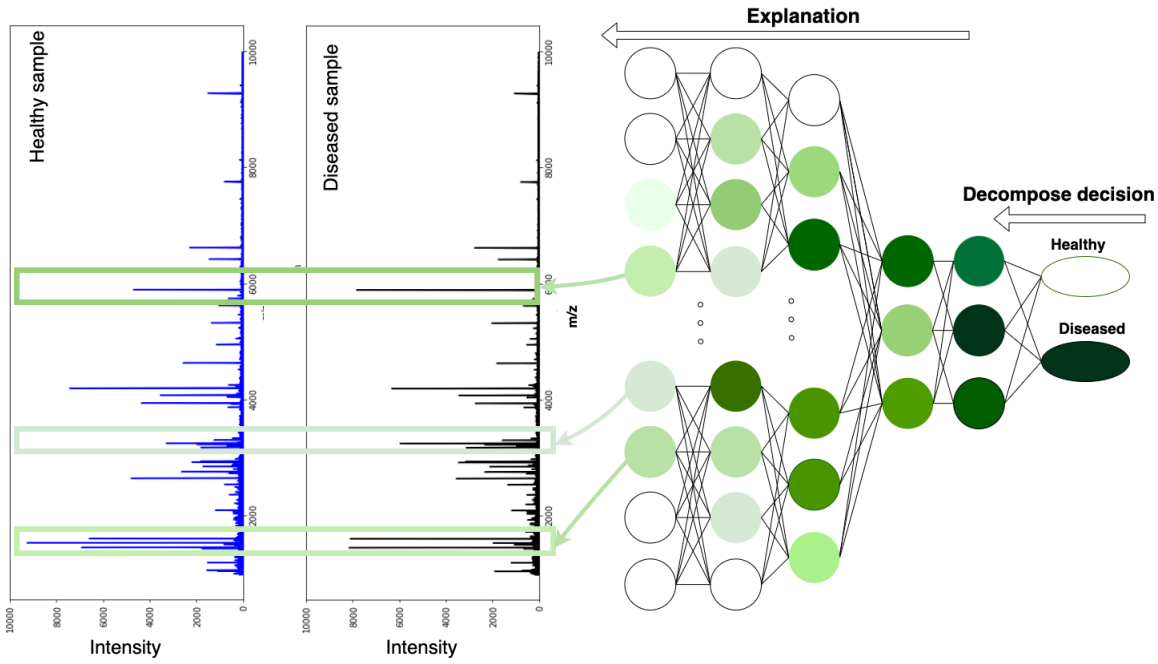


Figure 3.3: LRP illustration for proteomics data classification. This illustration shows the interpretation of a diseased sample (the spectrum in black) classification prediction. In LRP, the decision is decomposed layer by layer through backpropagation to the input layer. The relevance values of the input layer indicate which dimensions and to what extent make a role in the prediction. The bars show the prominent dimensions (features). In comparison with a healthy sample (the spectrum in blue), it can be realized that the network relies on differently abundant peaks to distinguish the diseased sample from the healthy ones.

model on detecting these biomarkers using a variant of LRP to interpret the DL model. Besides, other methods of interpretation including input gradient (grad), integrated gradient (int_grad), guided back-propagation (guided), deconvenet (dcnv), and smooth grad (smgrad) introduced in Section 3.2.2 are investigated. Further, the model is applied on a real-world public dataset, in which the selected biomarkers are compared with the state-of-the-art methods.

3.6.1 Spiked Dataset

We evaluate our proposed method on a public dataset known as *spiked data* (Fiedler et al., 2009; Kratzsch et al., 2005) and compare the effect of employing different interpretation methods in our pipeline. The spiked dataset contains proteomics mass spectra of control and case groups from human blood samples. The case group has been spiked with a protein mix of different concentrations. The amplitudes of 6 spiked peaks differentiate the spectra into case and control, and their known m/z (position) values can be used as ground-truth (Conrad et al., 2006). Thus, the main aim of this part is to investigate how well an algorithm can detect the m/z positions of the known six individual spiked peeks among all 42381 dimensions. The data contains 95 samples of 50 cases and 45 control spectra. The experiments are carried out on two concentration levels, 12.21nMol/L and 0.76nMol/L, referred to as *spiked160* and *spiked80*.

The results of our approach, i.e., the selected spiked peaks, are shown in table 3.2 and 3.3. The reported peaks are the closest to the spiked peaks ground-truth among almost 30 high-ranked features. From these two tables, we can see that LRP variants (attribution method), inp×grad, and int_grad are far more capable than signal (grad and smoothgrad) and function (guided and dCN) methods. It can also be seen from the results that, while there is no considerable difference between the variant of LRP in this application, one small peak (m/z 3149) can only be detected using LRP.z. The reason could be that since only LRP.z do not allow the fellow of negative values, this small peak is detected earlier. To better understand the negative values or contracting evidence, we run a systematic experiment on synthetically generated data in the next chapter in Section 4.4.2.

Prior to feature selection using the described DNN classification analyzer, the network should become generalized enough to allow the application of interpretation methods. This is what we addressed with transfer learning for the cases when only

a few labeled samples are available to train a DNN. In this situation, a simulated dataset of 5000 samples (Gibb and Strimmer, 2012) is fed to the network. The dataset contains two equal-size groups of spectra as control and case. Each simulated spectrum has more than 40 thousand mass values, as the real data spectra have. In addition, each one has 412 peaks, of which 24 are discriminating. They are equally spread in two groups and are set in fixed positions throughout the entire dataset. After training, the network re-trained on a real-world dataset of 81 samples and then fine-tuned on spiked data. This way, initializing the network weights should lead to better results since it is less likely that the optimizer gets trapped in a bad local minimum.

We observe from training the network that, while the objective function cannot converge on some subsets of samples, the pre-trained network can avoid that. Pre-trained weights lead to a more robust network that resulted in 97.1% ($CI \pm 2.68$) and 96.5% ($CI \pm 3.6$) generalization accuracies on spiked160 and spiked80, respectively. The seemingly large confidence intervals (CI) result from misclassifying one sample on different subsamples during training. Iterating training (train and validation) on 90% of randomly selected spiked160 (95 samples) and inferring on the rest, each time, leads to 100% or 88% testing accuracies. This means when the network performs 88% on testing, one spectrum out of nine ones was misclassified.

Table 3.2: Comparison of detected spiked peaks using nine interpretation methods on spiked160 data. We compare which spiked-in features are highlighted as the top 30 high ranked features with these methods using our pipeline. It can be clearly seen that the LRP variants, inp×grad, and int_grad in attribution category are far more capable than signal (grad and smoothgrad) and function (guided and dCN) methods.

peaks	grad	LRP.z	LRP.$\alpha\beta$	LRP.ϵ	inp×grd	`int_grad`	guided	dCN	smoothgrad
1047.20	-	**1047.91**	1046.76	**1047.91**	**1047.91**	**1047.91**	-	-	-
1297.51	-	1300.67	**1298.23**	1300.67	1300.67	1300.67	-	-	-
1620.88	1623.6	1621.91	**1620.48**	1621.91	1621.91	1621.91	-	-	1623.6
2466.73	-	2467.63	**2466.51**	2467.63	2467.63	2467.63	2463.63	-	-
3149.61	-	-*	-	-	-	-	-	-	-
5734.56**	-	-	-	-	-	-	-	-	-

* Although m/z 3149 is not selected as the top high-ranked feature because of its insignificant peak in comparison to larger peaks in the spectra (as illustrated in Fig. 3.4), it is selected as the 94th feature with our method using LRP.z. The other LRP rules can also select this peak but later as the less important feature. However, inp×grad and int_grad could not find this small peak. This is the reason we analyzed the noisy P.CA data and the visualizations by adopting the LPR.z rule.

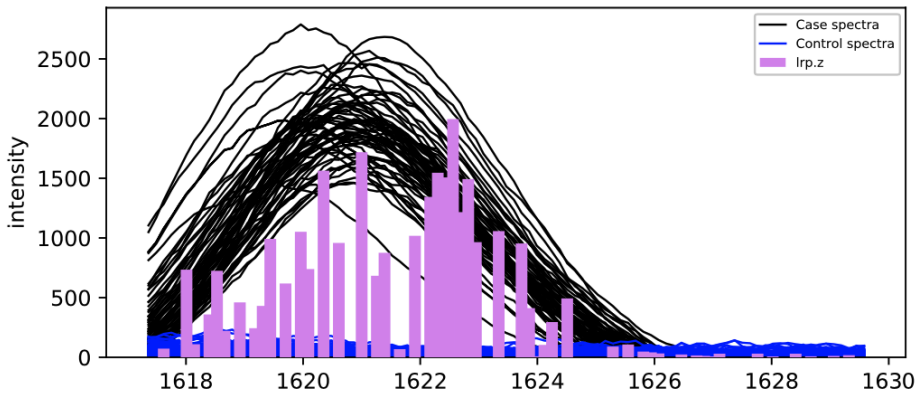
** The mean height of the signal in this peak is less than 40 that is comparable to the level of noise in both spiked160 and spiked80 datasets (Conrad et al., 2017). Therefore, this peak cannot be selected as a discriminating feature.

Table 3.3: Comparison of detected spiked peaks using the nine interpretation methods on spiked80 as the top 30 high ranked features. Similar to results on spike160 data the LRP variants, inp×grad, and int_grad in attribution category are far more capable than signal (grad and smoothgrad) and function (guided and dCN) methods in our application.

peaks	grad	LRP.z	LRP.$\alpha\beta$	LRP.ϵ	inp×grd	`int_grad`	guided	deCN	smoothgrad
1047.20	-	1040.61	1041.76	1040.61	1040.61	1040.61	-	-	-
1297.51	-	1298.35	**1298.0**	1298.35	1298.35	1298.35	-	-	-
1620.88	-	**1620.87**	1619.7	**1620.87**	**1620.87**	**1620.87**	-	-	-
2466.73	-	**2467.63**	2468.6	**2467.63**	**2467.63**	**2467.63**	-	-	-
3149.61	**3151.25**	-	-	-	-	-	-	-	**3151.25**
5734.56	-	-	-	-	-	-	-	-	-

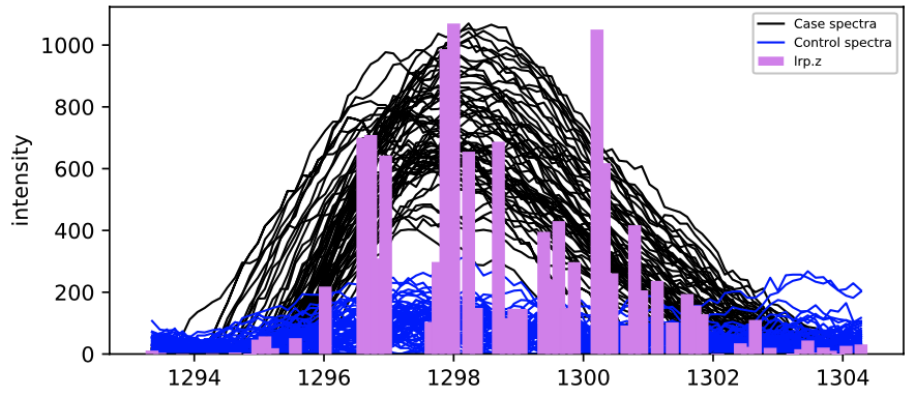
We further explain the results in Fig. 3.4 by visualizing the output of one of the interpretation methods. The figure shows the mean of the normalized LRP.z values of a spiked160 spectrum overlaid on the distribution of case and control spectra of the dataset around the selected spiked peaks. The visualization around the spiked peaks, as shown in these plots, indicates the wide peak range that causes the deviation on the decided features from the spiked ground truth peaks in tables 3.2 and 3.3.

The spiked peaks among the top 30 selected features using our pipeline are supposed to be selected as the most discriminating features. However, in Figure 3.5 we illustrate that the selected features that are ranked better than the true spiked peaks are more discriminating. For example, it is apparent from the plot that the intensity gap between the case and control samples around feature 1021 is larger than the corresponding intensity gap around feature 1047.

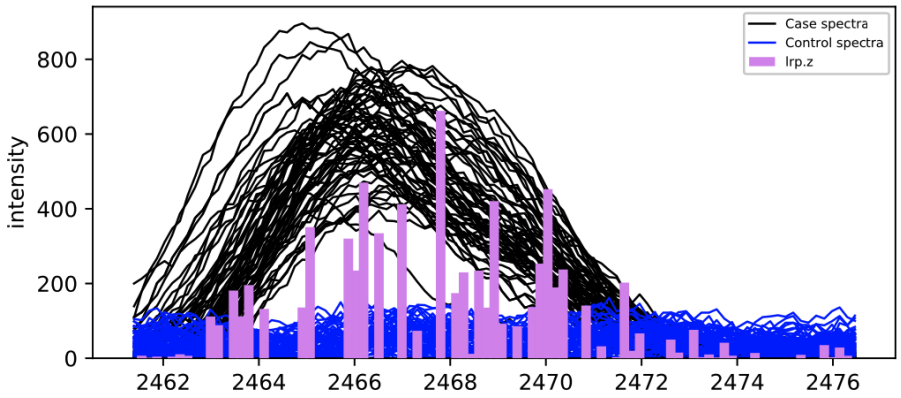


(a) m/z 1620.88

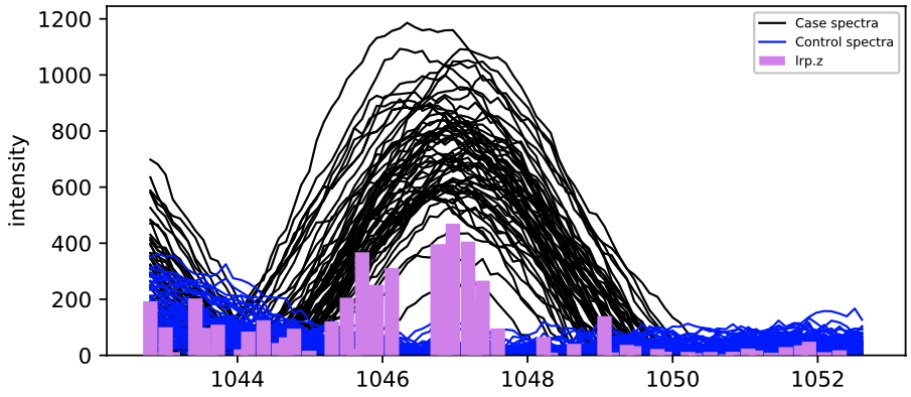
See the next pages for a detailed description.



(b) m/z 1297.51

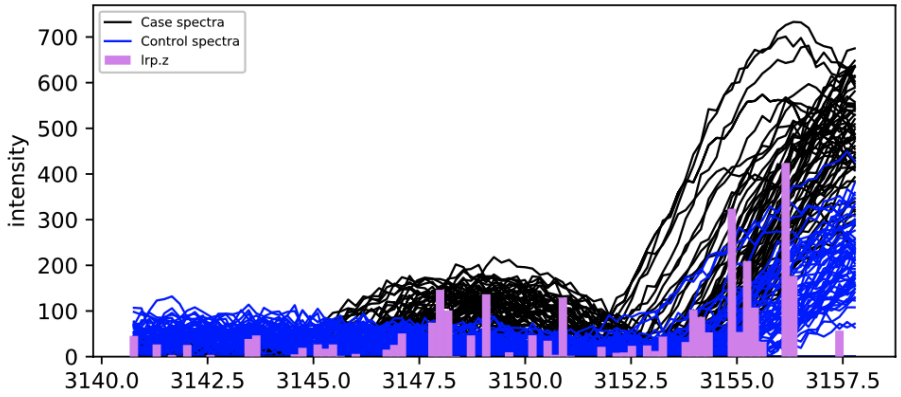


(c) m/z 24666.73



(d) m/z 1047.20

See the next page for a detailed description.



(e) m/z 3149.61

Figure 3.4: Visualization of the relevance values around the spiked peaks. Black and blue show the diseased and healthy spectrum of spiked160, and the bars are the average of the normalized LRP.z values over the entire samples. The bars are scaled to the maximum intensity of the spectrum. x, y axes represent the intensity and m/z value of the spectrum. This visualization shows that the network heavily relies on discriminating areas to make classification predictions. Figure (e) demonstrates that the network still recognizes the very small differentially abundant peak. But, it can not be detected among the first 30 differentially abundant peaks since the level of signal is close to level of the noise.

Therefore, the DNN tends to rely more on these areas to make the classification decision. We can also learn from this plot that not only the individual features are essential for the DNN to make a classification decision, but a Gaussian range around high-ranked ones also plays a crucial role. For example, relevance values around the m/z 1021 are considerably higher than the relevance value of individual m/z 1047. Therefore, we cannot expect a DNN to classify the two groups based on only individual features.

3.6.2 Pancreas Cancer Dataset

The Pancreas Cancer dataset (P. CA) is another publicly available dataset (Fiedler et al., 2009), where only the health status of instances are known as labels. It contains 81 spectra having 42391 features collected from pancreatic cancer patients and healthy control patients. To demonstrate the performance of our model, we report the features that are selected as discriminating features using our method and compare them with the ones reported by other methods on this benchmark

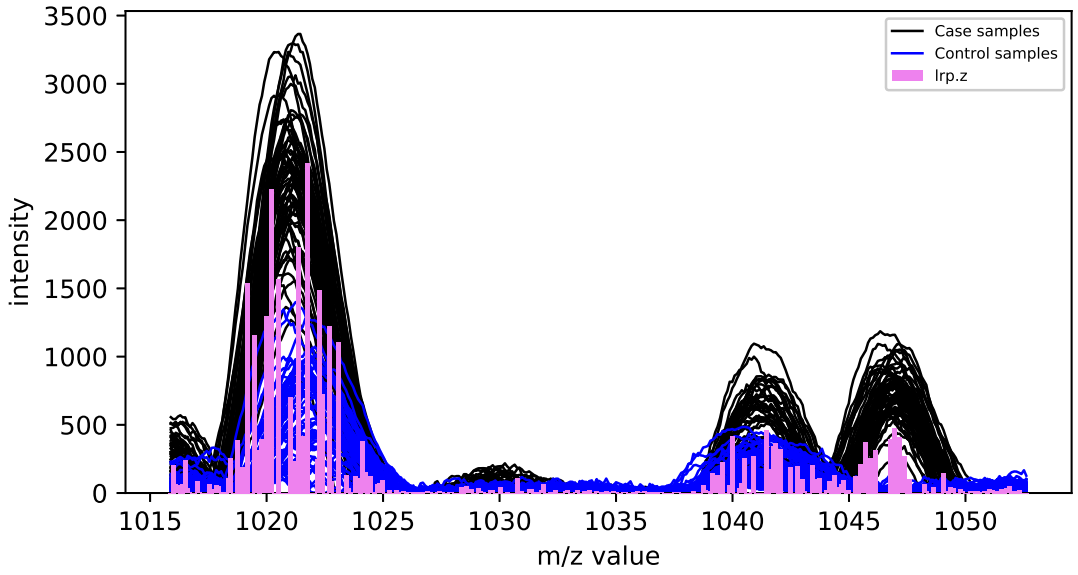


Figure 3.5: Visualization comparison of two selected peaks. This plot illustrates the selected spiked 1047 in a wider range to include the selected feature 1021. This illustration shows that m/z 1021 is selected prior to the ground truth m/z 1047 since the network sees larger differences between the two classes. Black and blue show the diseased and healthy spectrum of spiked160, and the bars are the average of the normalizer LRP.z values over the entire samples. The bars are scaled to the maximum intensity of the spectrum.

dataset.

As described previously, due to the lack of sufficient training samples on the public dataset for training a deep network, we initialized the weight of the classification network with the representation of simulated data. We achieved 98%-95% training-testing average accuracy, while almost all the structures of DNN we tried from shallow to deep and narrow to wide could not become generalized correctly. The classification decision is interpreted using LRP.z rule to extract the important parts. Fig. 3.6 illustrates the average of normalized LRP.z over the entire dataset, around two of the high-ranked features. The relevance values are overlaid on top of the mean of the case and control spectra. These two features are illustrated due to the large impact on the classification decision after feature selection (see Fig. 4.4).

We compare our feature selection method with benchmark methods on this dataset. A BinDA-algorithm-based method (Gibb and Strimmer, 2015) reported 30 peaks

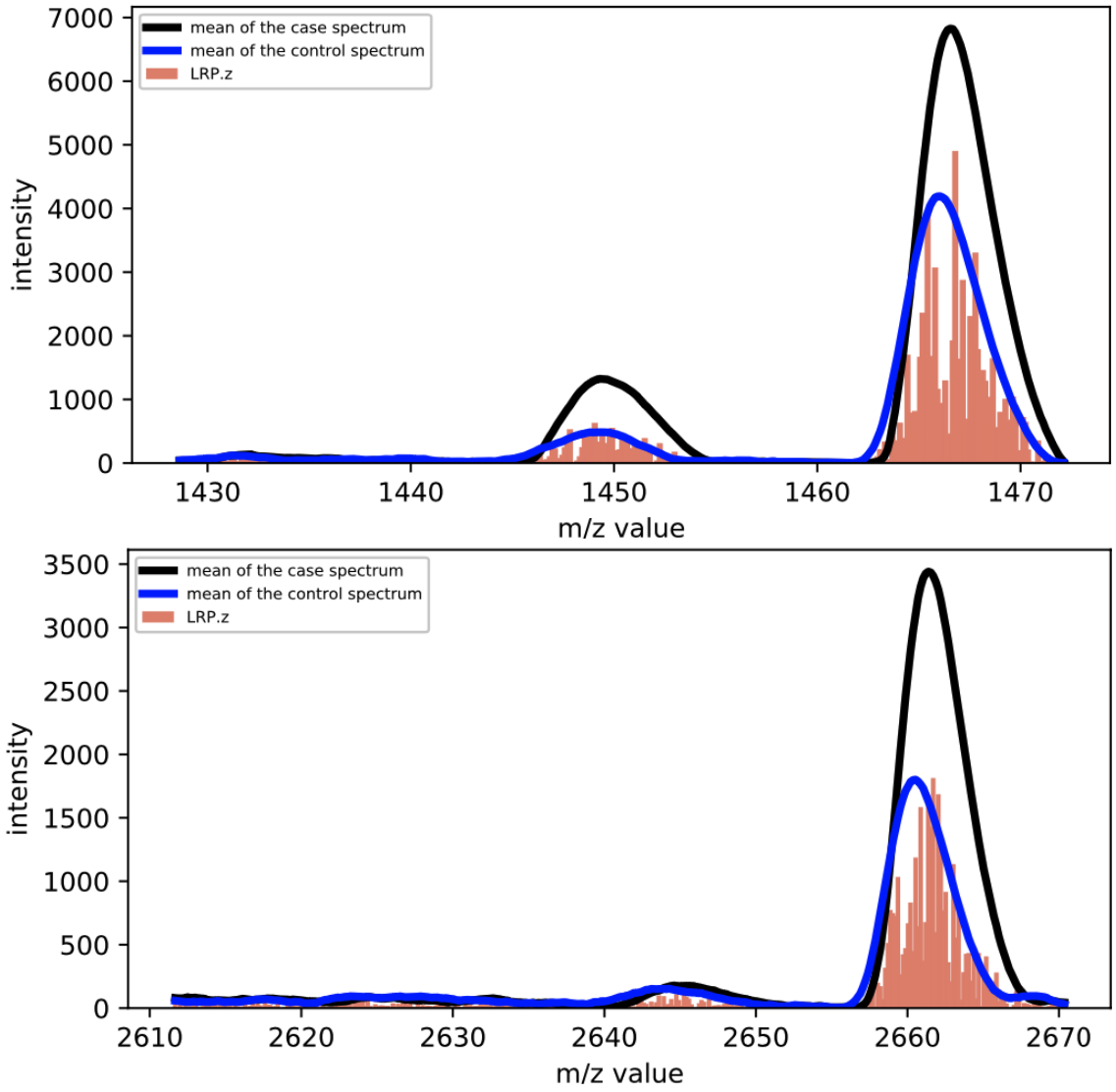


Figure 3.6: Illustration of the relevance values around the second (m/z 1465) and forth (m/z 2661) high ranked features of P.CA data. These features are picked for illustration due to their largest impact on the classification accuracy after feature selection. The means of the case and healthy spectrum are shown in black and blue, respectively. This illustration elucidates that the network heavily relies on differentially abundant peaks on real data.

m/z 4495, 8868, 8989, 1855, 4468, 8937, 2023, 1866, 5864, 5946, 1780, 2093, **5906**, 5960, 8131, **1207**, 4236, **2953**, 9181, 1021, **1466**, **4092**, 4251, 5005, 8184, 1897, **3264**, 2756, 6051, 1264, and m/z 8937 as the most discriminating features for pancreatic progenitor cell differentiation. Note that, the bold m/z values indicate the features that are also discovered by our method.

Using a compressed sensing-based approach, Conrad et al. (2017) identify peaks with m/z **1464**, **1546**, **1944**, **5904**, **1619**, **4209**, and m/z **2662** as discriminating features, which are all selected with our approach.

Using our method, peaks with m/z values **4212.36**, **1465.43**, **3264.36**, **2661.37**, **5909.96**, **4092.18**, **1616.98**, **1545.91**, 4647.56, 6636.87, 3191.41, 2934.34, 5338.51, **2953.42**, 1060.26, and m/z 3242.47 are ranked as the most discriminating features to achieve the state-of-the-art classification accuracy of 95% (Conrad et al., 2017). The first eight selected peaks in our approach have been selected with at least one of the earlier approaches. The mass shift of 1 to 3 Dalton on the m/z axis among the identified peaks over different study is likely arising from different preprocessing and postprocessing procedures.

3.7 Extension to 2D Proteomics Data

In the preceding sections, we introduced our new feature selection method on high-dimensional vector-shaped data and showed the efficiency and robustness of the proposed method with the case study of mass-spectrometry proteomics data. In this section, we extend our feature selection methodology for data of matrix form. The extension is formulated to apply more complex proteomics data than mass spectra data. Oftentimes, quantifying complex proteomics samples in just a single mass spectrum is impossible. This is because different peptides may have similar masses after ionization in mass spectrometry, and on a single quantified spectrum, these peptides are overlaid, making it impossible to differentiate those peptides. One possible way to simplify the quantification is to add one more separation level prior to mass spectrometry, which is liquid chromatography (LC) separation in our case.

LC separates peptides based on their chemical affinity in specific retention time, and a mass spectrometer quantifies them afterwards. This separation enables the dissimilar ions with similar masses to enter the mass spectrometer at different times, which adds the time variable to the quantified spectrum. Stacking all the spectra

through a time axis then acquiring LC-MS map whose x, y, and z axes present the m/z, RT, and intensity of ions, respectively. LC-MS data acquisition and interpretation have been introduced in more detail in Section 2.1.4. LC-MS allows the analysis of complex biological mixtures, such as body fluids (e.g., blood or urine). Due to the precise and fast quantification process, it is widely used in high-throughput proteomics applications (Wang et al., 2016; Hoffmann et al., 2019; Souza et al., 2017), such as disease diagnosis (or prognosis), biomarker detection, or drug target identification. The main goal is to select discriminating m/z, Rt pairs between LC-MS maps of healthy and disease samples. Ions' masses, the time they are released from the chromatography column, and the intensity of ions are used as the specification of peptides, which are searched in databases to be identified. The identified peptides indicate the possible disease fingerprint. An overview of the feature selection model for LC-MS data is shown in figure 3.7.

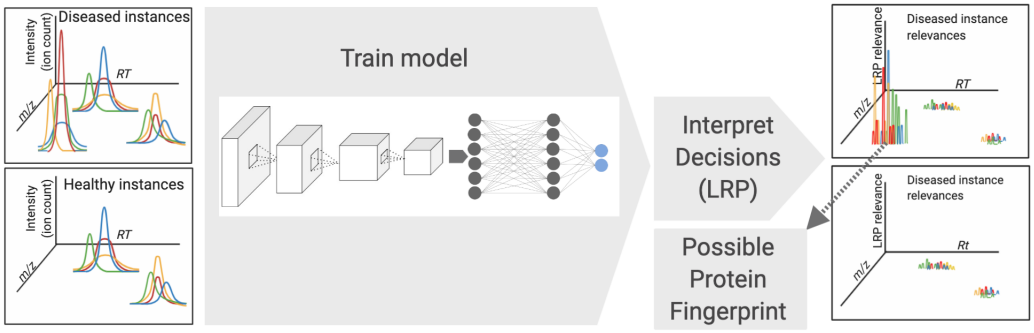


Figure 3.7: Overview of our approach for discovery of disease related biomarkers. A CNN is robustly trained on diseased and healthy LC-MS maps for the binary classification task. The predictions of the trained network are interpreted by layer-wise relevance propagation strategy on samples belonging to each class, separately. The peaks on the interpretation heatmap represent the peaks that the CNN relies to make classification decisions. The statistically significant peaks that occurred only on heatmaps of diseased samples are considered as possible diseased biomarkers.

Similar to MALDI-MS feature selection, we formally formulate the LC-MS feature selection as follows: let $I_n \in \mathbb{R}^2$ for $n = 1, ..., N$ be a series of LC-MS maps, which take $O_n \in \{0, 1\}$ as the medical condition labels. Each (x, y) pair of I, where $x = m/z$ and $y = RT$, contains ion-count demonstrating features on LC-MS map. The aim is to find the smallest subset of $(\hat{x}, \hat{y})$ pairs whose ion-counts are differentially abundant between conditions 0 and 1. Similar to MS biomarker detection, the strategy is to design a network, modeled as function f, to classify LC-MS samples into two classes,

and learn from the prediction behavior to detect $(\hat{x}, \hat{y})$ pairs using LRP. A CNN with L layers is trained on LC-MS instances. By assuming that the network is generalized accurately, predictions are redistributed backward layer by layer to give a score to all the input features. A feature $(\hat{x}, \hat{y})$ will be attributed strong relevance if the function f is sensitive to the presence of that feature. The relevance value of all (x, y) pairs form the matrix of relevances, R_i^1, known as a heatmap. The goal is to employ this information to verify the predicted medical condition and find the most relevant attributions associated with this condition.

The first step again is to design a robust classifier, a CNN, to classify the LC-MS samples of two classes that we are interested in their differences. We train networks with different widths and depths from standard structures like variants of ResNet (He et al., 2016b) to customized structures. We observe that training very deep networks like ResNet32 on the LC-MS data (both synthetic and real data) leads to overfitting, while a network with a few layers fits with high accuracy. The outperformance of the customize network over very deep networks can intuitively be explained by the local dependent characterization of the peaks on the LC-MS map. Very deep networks capture both the local dependencies gained by the reach feature representation and the global dependencies gained by the large receptive field. Therefore, very deep networks may learn some global patterns irrelevant to the data information but relevant to the noise, such as quantification calibration error in data acquisition. This may arise from the insufficient amount of training data with respect to the number of parameters, increased by the depth of the network.

More strategies we take for generalization are delayed to chapter 4, where we introduce how we take advantage of synthetically generated data to tune the parameter and hyperparameters of the network for convergence and generalization assurance.

Once the network has been robustly trained, we employed the network to learn the representation of the real data and to discover the discriminating peaks from its interpretation. It is worth mentioning that our assumptions to use the interpretation for biomarker discovery are the reproducibility and robustness of the interpretations, which will be discussed and justified in the next chapter, Section 4.4.3. In Section 3.6 we showed that LRP results in a better outcome for MS data than other neural networks interpretation approaches. Since the LC-MS is inherently similar to MS and basically is the stack of many MS spectra, we also interpret the CNN's decision here with LRP given by Eq (3.30). This time relevance values in the input layer R^1 build a heatmap over the matrix of instances, likewise the heatmap for images. R^1

which is calculated for all pairs of $(m/z, RT)$ demonstrates how much index (x, y) representing m/z and RT contributes to the decision-making.

Figure 3.8 depicts the interpretation heatmap on local area of a peak that only occur in diseased LC-MS samples.

Considering offsets, the presence of noise, and different peak indices of instances, we interpret the decisions on statistics of the whole training-set. We take the mean of LC-MS samples belonging to the diseased and healthy classes separately. Each mean is given to the trained network f, and the predictions are interpreted by the LRP function. This results in two matrices of diseased relevance values R^1_{dis} and healthy group's relevance values, R^1_{hel}.

$$R^1_{dis} = \text{LRP}(f(\frac{1}{N_{dis}} \sum_{n \in \text{dis}} I_n)),\ R^1_{hel} = \text{LRP}(f(\frac{1}{N_{hel}} \sum_{n \in \text{hel}} I_n)),$$

where N_{dis} and N_{hel} are the number of samples in diseased and healthy classes, respectively. The spatial location of peaks on the LC-MS map are widely distributed, and the exact location of peaks can be estimated by finding the index with maximum intensity in a predefined window. The size of the window is estimated through a statistic on the length of peaks along m/z and RT axis. To this end, similar to the MS feature selection, we first select the peak with the strongest relevancees from R^1_{dis}. Then, the neighbor's relevances in the window are zeroed. We iterate this process until all the high-intensity relevances are covered. The selected peaks are distinguished as biomarkers if corresponding indices on R^1_{hel} are attributed non-negative relevances.

To extract the biomarker from an unknown sample, the sample is fed to the network to be classified. The classification prediction is interpreted using LRP, similar to the training phase. The high relevances are selected locally from LRP interpretation, similar to selecting the peaks from training samples. These peaks, corresponding to the high relevances, are distinguished as biomarkers if their indices attributed strong on R^1_{dis} and non-negative on R^1_{hel}.

3.8 Results on LC-MS Data

In this section, the performance of the proposed method is assessed on a published benchmark LC-MS data, (Tuli et al., 2012) which we refer to as real data. Many

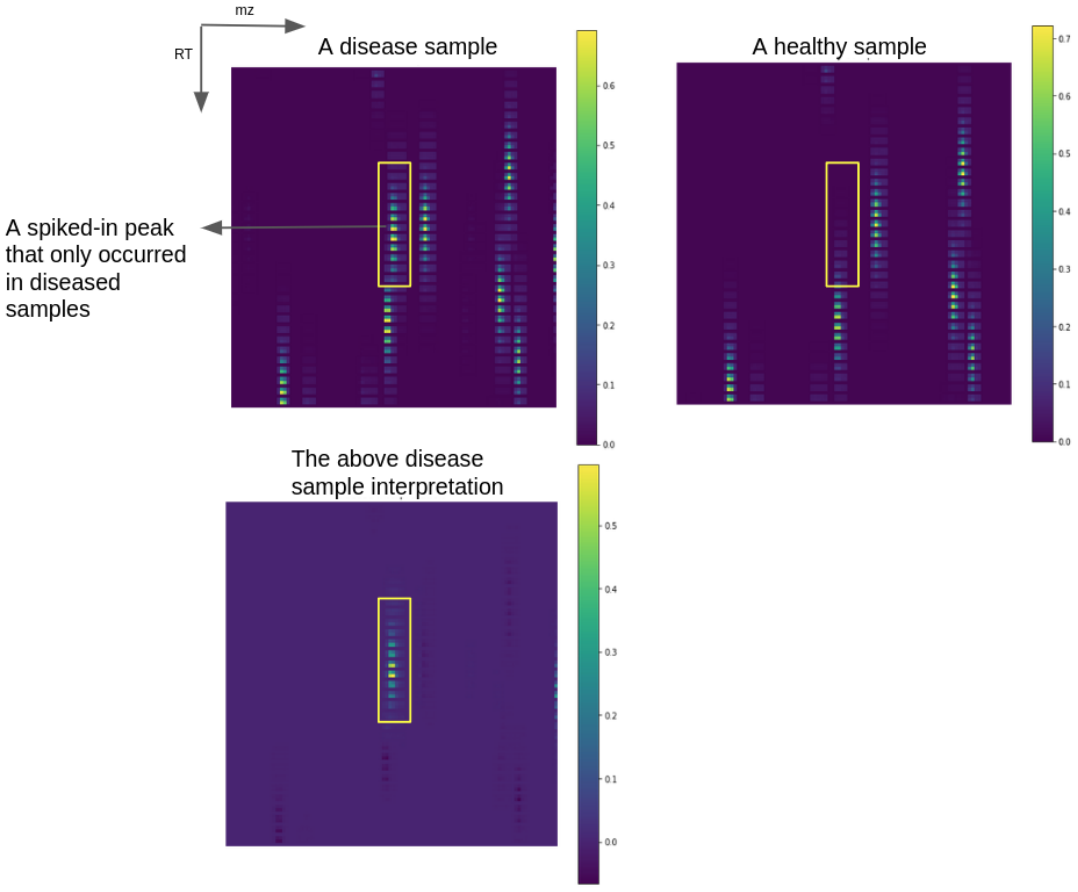


Figure 3.8: Visualization of real LC-MS data and its prediction interpretation using LRP in the local area, where peaks spike only on disease samples. As it is apparent, the discriminating peak in the disease sample is captured by the network's prediction interpretation of this sample. Despite widely used gradient based methods that are locally calculated, LRP takes into account of the whole input, which makes it less prone to the discontinuity problem (Montavon et al., 2018) and consequently more applicable for very noisy proteomics data.

Table 3.4: Specification of the real data spike-in peptides. Base peak chromatograms of the group with spike-in peptides are presented based on their mass-to-charge ratio (m/z), retention time (RT), and ion charge.

Features No.	1	2	3a	3b	4	5a	5b	6	7a	7b	8	9a	9b
m/z	501.25	450.23	530.78	354.19	523.77	648.84	432.89	586.98	624.99	630.35	943.43	712.43	570.15
Charge	2	2	2	3	2	2	3	3	3	3	3	4	5
RT(min) start-end	4-8	45-49	53-56	53-56	59-62	63-67	63-67	73-77	77-81	82-86	79-83	103- 107	103-107

other Mass spectrometry datasets are available at repositories such as PRIDE or CompMS. However, the focus of this section is to assess the feature selection on a raw LC-MS map of samples from two conditions (healthy and control) with known biomarkers presented by their m/z and RT, which is perfectly met in the selected dataset. The known biomarkers are not considered for training because our method is needless of labels at the biomarker levels, but we used them for the assessment of the method. All the parameters and hyperparameters of the model classification network are tuned on the synthetically generated data, which are delayed to Chapter 4.

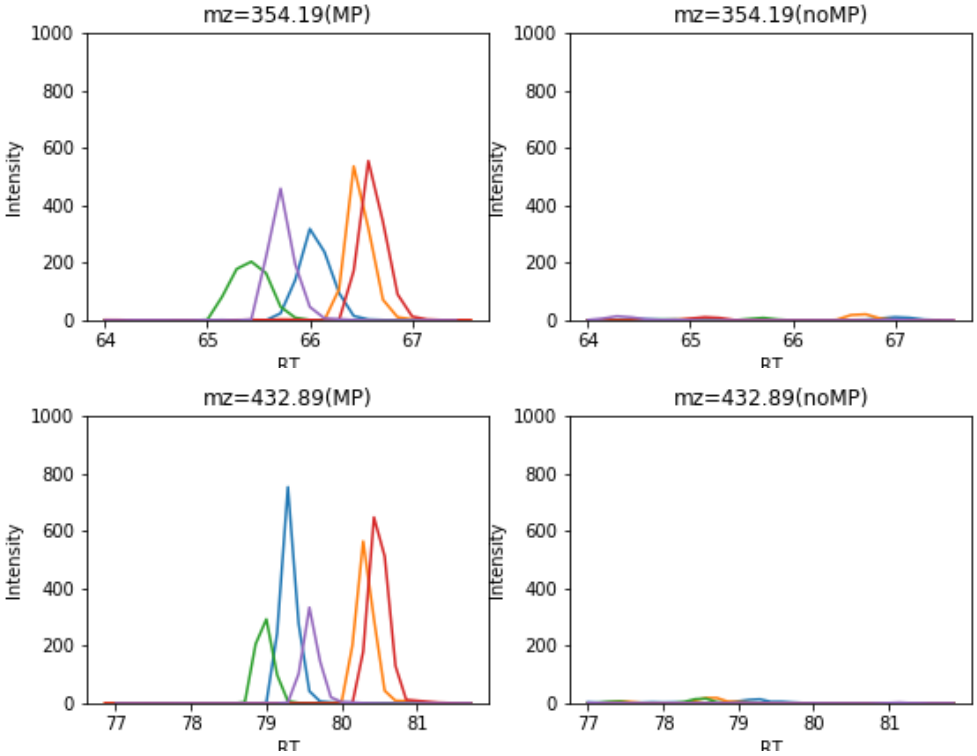
3.8.1 Real Data

The real LC-MS dataset consists of two groups. The first group was derived from five serum samples of healthy individuals spiked with a known concentration of spike-in peptides. The second group was obtained from the serum samples only. We refer to the first and second groups as diseased and healthy, respectively. The added peptides to the diseased group are the selection of nine peptides with different concentrations to be representative of real datasets. They have predictable retention behavior and elution order that let the ground truth available in m/z and RT (Tuli et al., 2012). LC-MS acquisition yields 13 peaks from nine peptides due to the different charges. The specifications of these peaks are presented in Table 3.4.

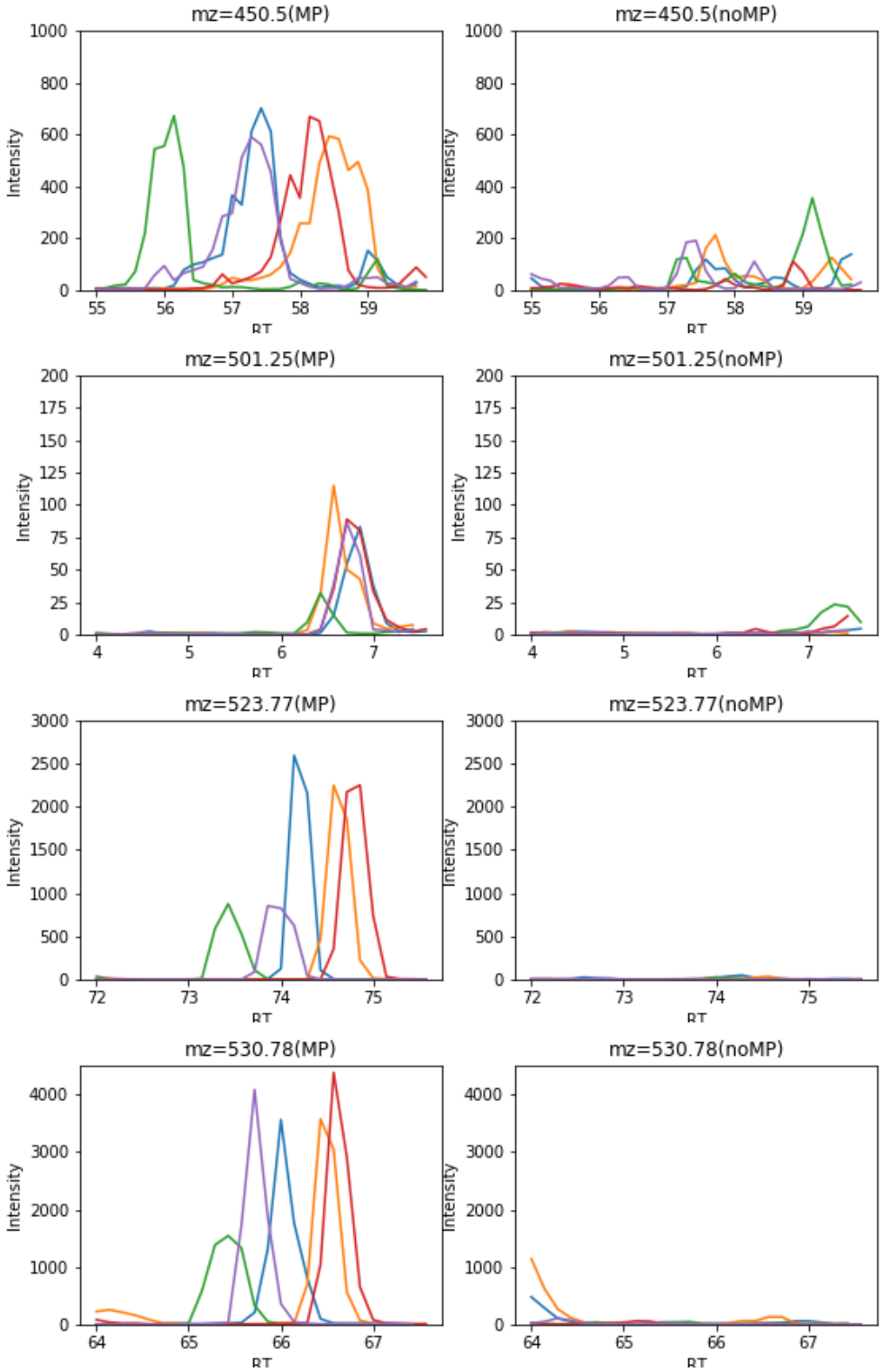
Figure 3.9 provides the visualization of the spiked-in peaks in this dataset. The chromatograms are zoomed into each of the 13 unique features of serum with spike-in peptides and serum alone groups. As is apparent from the visualization, peaks are spread in different concentrations with different mean and variances to be representative of the real mass spectrum data.

The proposed method is intended to detect differentially abundant spike peaks as biomarkers and keep detected FP peaks low. Therefore, the evaluation will be reported as the exact number of TP and FP peaks. We quantize the raw data and form chromatograms matrices. This outcome is then converted into images whose

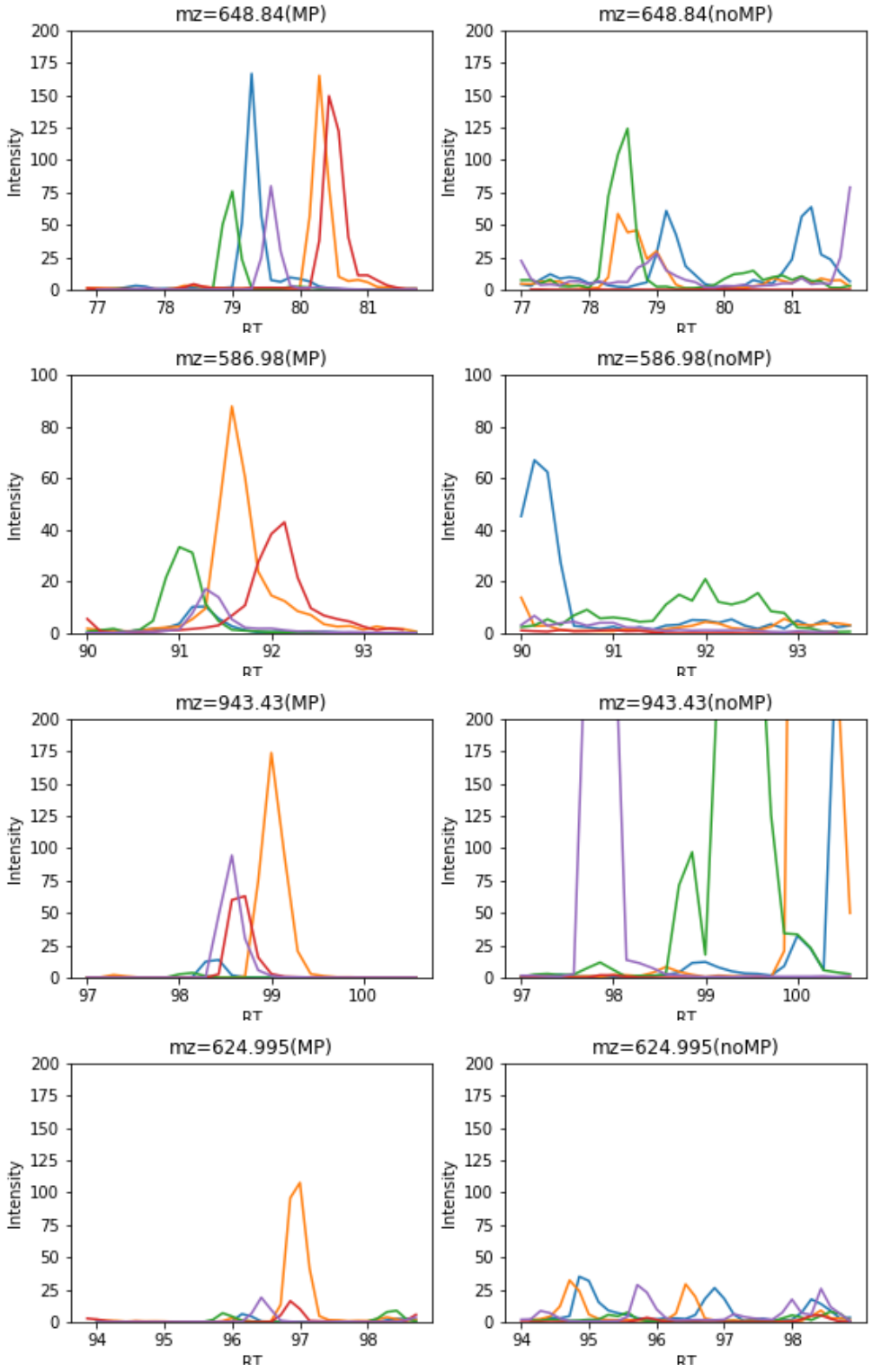
width and height are m/z and RT, respectively. Each RT bin on the y-axis presents seven seconds of the MS level-1 scan, and the x-axis covers ions of m/z 350 to m/z 2000. Pixel intensities demonstrate the ion counts. LC runs for 240 minutes; however, similar to the benchmark methods, we filter the samples to retain features within 150 minutes because there is no significant peak out of this range. Besides, we remove the features with the ion-count intensities less than two as the only noise reduction on the samples.



See the next pages for the remaining plots and a detailed description.



See the next pages for the remaining plots and a detailed description.



See the next pages for the remaining plots and a detailed description.

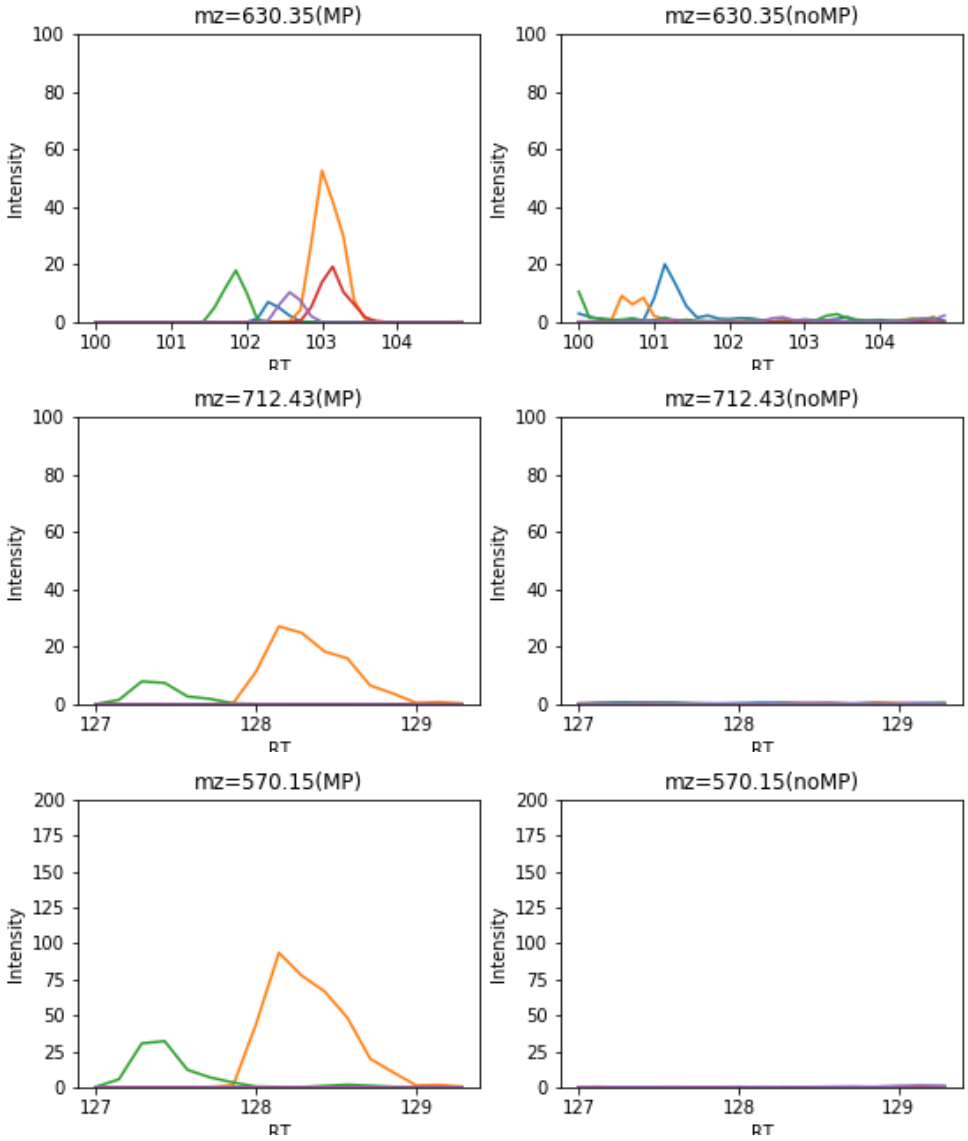


Figure 3.9: Visualization of all 13 spike-in peptides in the real LC-MS data. Chromatograms are zoomed into the location of spiked-in peptides in the group of serum samples mixed with the spiked-in peptides (MP) and group of serum samples only (noMP). Different colors show the distribution of different serum samples. As it is apparent, different samples contain different concentrations of spike-in peptides, such as m/z 432.89, 523.77, and m/z 648.84. Besides, some peaks are not spiked on all serum samples, including m/z 712.43 and m/z 570.15.

3.8.2 Results

This section describes the comparison results of our method with four other LC-MS analysis workflows. A false discovery rate based on statistical analysis is applied to the output of these workflows to reduce the number of selected false positive peaks. We first introduce this statistical analysis and then present the comparison results.

False Discovery Rate

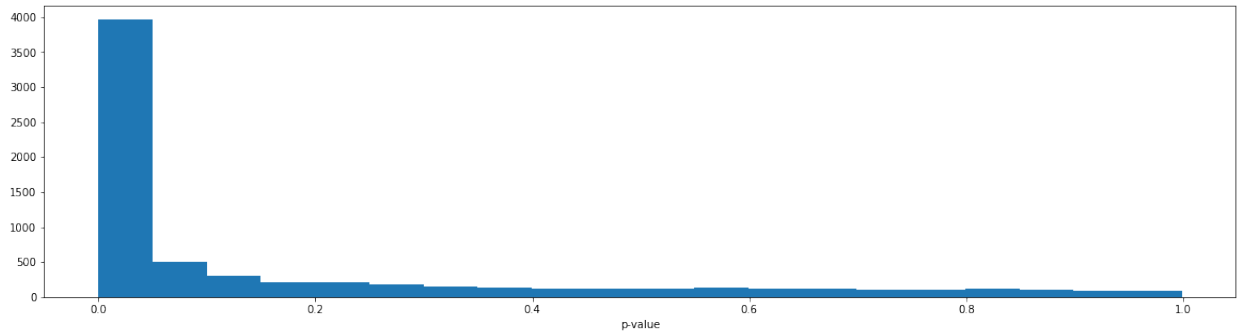
A false discovery rate is a tool commonly used in high-throughput sequencing to weed out, for example, dimensions of the data that are wrongly selected as significant. To this end, a statistical test is calculated for all selected features to obtain the distribution of the p-values. The p-values in our problem indicate the significance of the variables for discriminating two groups of healthy and diseased. The smaller the p-values, the stronger the evidence to reject the null hypothesis, which says given the values of variables in samples of two groups, no real difference existed; therefore, it can be concluded that the difference in group abundance is significant. The p-values smaller than 0.05 are usually considered significant. This means there is a 5% chance of getting a significant difference where no difference exists in the group means. 5% is acceptable for one test. But the multiple testing of thousands of components, common in omics experiments, can result in a large number of false positives. To address this problem, most attempts are towards adjusting p-values to a more resealable value.

As we form the histogram of p-values on the output of selected features, we expect the p-values to be heavily skewed and closer to zero. This is depicted in Figure 3.10. It is shown that around 4000 features are significant, with p-values smaller than 0.05. To control the false discovery rate or to limit the number of false positives, Benjamini-Hochberg punishes p-values accordingly to their ranking:

$$\text{q-value} = \text{p-value} \times \left(\frac{\text{total number of p-values}}{\text{p-value rank}}\right)$$

The q-value is the name given to the adjusted p-value. By adjusting the p-values, some selected features will be no longer considered statistically significant. For instance, p-values of 0.04 turn into 0.06 after the false discovery correction, which is no longer significant. Applying the p-value correction following the feature selection workflows reduces the false positives, as reported in Table 3.5. Note that q-values

Figure 3.10: Density distribution of p-values on all selected features from our feature selection method. It can be seen that most features are skewed to the left side of the p-value histogram, which indicates most of the selected features are statistically significant. To limit the false positives, we apply the p-value correction following the feature selection workflow.



will not always result in fewer false positives, but it can be perceived that it gives a more accurate indication of the level of false positives for a given cut-off value. Besides, it is not guaranteed to not lose the true positives with the false-discovery correction. For instance, as it is shown in Table 3.5, although the false-positive rates reduce significantly, all the workflows, including MZmin 2, XCMS, and DLearnMS lose the true positive peaks after the p-value correction.

T-test

The t-test in the statistic is originally developed to deal with small samples by introducing the t-distribution when: (1) underlying sample distribution is normal, (2) population standard deviation is unknown, and (3) sample size is too small to apply central limit theory (which says if the sample size gets large enough we can just use a normal distribution for our sample statistics). In this case, for hypothesis testing, the sample standard deviation is used in place of population standard deviation for the test statistic. Different samples of small size from the population might have different standard deviations, which brings the need for t-distribution to adjust the additional uncertainty around the sample's standard distribution. In this section, according to the limitation to the number of samples in the real dataset, we use the t-test for multiple testing, as mentioned earlier.

Table 3.5 compares the feature selection of our proposed method on the described real dataset with the benchmark pipelines using msInspect, MZmine 2, Progenesis,

and XCMS. The first row in Table 3.5 demonstrates that our method outperforms the other pipelines in terms of detecting fewer FP peaks. Moreover, our analysis does not depend on the preprocessing steps used in other workflows. Nevertheless, LC-MS preprocessing approaches (Liu et al., 2020a; Kantz et al., 2019) could be potentially added to our DLearnMS framework and could even further improve the performance results. The t-test for $p < 0.05$ is calculated on each selected feature and multiple testing correction is applied. The features that satisfy $q < 0.05$ are selected as the discriminating features, presented on the third row of Table 3.5. The fourth row shows the number of selected features that satisfy $q < 0.05$ and the fold change (FC) > 10. We detect nine biomarker peaks similar to msInspect, while we achieve almost ten times fewer FP peaks, 195 in comparison with 2099 FP peaks in msInspect. We also outperform MZmine 2 and Progenesis with respect to both evaluation metrics, namely the number of biomarker peaks (seven in MZmine 2 and eight in Progenesis) and FP peaks (539 in MZmine 2 and 467 in Progenesis). Although XCMS achieves the best results with respect to the number of FP peaks, 66, which is the smallest number of FP peaks, its performance concerning the number of detected biomarker peaks, has dropped to seven. It should be noted that in high-stake decisions, we should always consider the FP and TP detection trade-off.

The biomarker peaks that are selected according to the statistical analysis are presented in Table 3.6. Six peaks that are commonly selected by all four other methods as differentially abundant peaks have also been detected by our method.

Implementation Setup

The experiments in this study are implemented in Python for data analysis, Scikit-learn library (Pedregosa et al., 2011) for ML analyzes, Keras (Chollet et al., 2015) with Tensorflow backend (Abadi et al., 2016) for DL analysis, and "iNNvestigate" library (Alber et al., 2019) for DL interpretation analysis on a machine with a 3.50 GHz Intel Xeon(R) E5-1650 v3 CPU and a GTX 1080 graphics card with 8 GiB GPU memory. We use the weight of the network that has been trained on synthetic data for initializing the network for training the real data. We retrain the whole network on the real data using leave-one-out cross validation. The classification network is trained for 20 epochs with batch size of two using Adam optimizer (Kingma and Ba, 2014) with the learning rate of 0.00001 and momentum of 0.9. We use binary cross-entropy as the loss function. The kernel size in all layers is set to 3×3 with the

Table 3.5: Feature selection comparison of the proposed method with MZmine 2 (Pluskal et al., 2010), Progenesis LC-MS (Qi et al., 2012), and XCMS (Smith et al., 2006), which are all presented in (Tuli et al., 2012). The total number of selected features is represented for all methods in the first row. The baseline methods report only used those features for statistical analysis, which are presented in at least two replicates in each group. The third and fourth rows demonstrate the number of features satisfying two representative criteria, including t-test with multiple hypothesis testing (q-value< 0.05) and fold change (FC > 10). The plus sign denotes the combination of different criteria. The numbers written in parentheses indicate the selected biomarker peaks. In compression with msInspect we achieve 10 times fewer false positive with comparable number of true positive peaks. We outperform Mzmine 2 and Prognesis by detecting more true positive and fewer false positive peaks. Although XCMS detects fewer false-positive peaks when applying the statistical analysis, it finds two fewer biomarkers than DLearnMS.

	msInspect	MZmine 2	Progenesis	XCMS	DLearnMS
# All selected features	31168 (12)	12271 (12)	9267 (9)	21486 (13)	8044 (12)
# Features for statistical analysis	6525 (9)	12092 (9)	8415 (9)	8703 (10)	8044 (12)
t-test ($q < 0.05$)	4824 (9)	3505 (7)	4465 (9)	1896 (7)	3985 (11)
t-test ($q < 0.05$) + FC (> 10)	2099 (9)	539 (7)	467 (8)	66 (7)	222 (9)

Table 3.6: Real data biomarker detection comparison according to the statistical analysis. Detected differential abundant spike-in peaks are shown by checkmarks. Note that our method detects all the features that are commonly selected by all other methods.

Features No.	1	2	3a	3b	4	5a	5b	6	7a	7b	8	9a	9b
msInspect	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	-	-	-
MZmine 2	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-
Progenesis	✓	✓	✓	✓	✓	-	✓	-	-	✓	-	-	✓
XCMS.	✓	✓	✓	✓	✓	-	✓	-	-	✓	-	-	
DLearnMS	✓	✓	✓	✓	✓	-	✓	-	-	✓	-	✓	✓

dropout rate of 0.3. The convolution layers in the network are two dimensions and contain the following number of kernels: 32 in the first and second layers, 64 in the third layer, and two in the fourth layer. The fully connected layer as the last layer has two neurons for binary classification.

3.9 Discussion

This chapter introduced a novel DL biomarker detection method for high-throughput data. Applying ML and DL on raw data without preprocessing could be challenging due to the sparsity, small sample size, complexity, and high noise level. Despite available tools, current workflows require several steps of data preparation. Besides, interpretation of ML and DL models is often neglected in favor of precision, despite the importance of decision explanation in biomedical settings. We developed a DL method backed by interpretation strategies to address the aforementioned issues. We designed and trained a DL classification network to learn the representation of data consisting of instances of healthy and diseased individuals and interpret the network decisions to learn what brings the network to certain decisions.

We demonstrated that our biomarker detection approach achieved better performance on a benchmark LC-MS dataset than conventional methods in detecting fewer false positive (FP) peaks and more true peaks as biomarkers despite being independent of otherwise necessary preprocessing steps. Based on the data presented here, our experiments showed that although one of the methods, XCMS, finds fewer FP peaks, it loses low-intensity markers (9a and 9b). However, it should be noted that FP reduction should avoid losing TPs. Especially in medical domain applications, these low-intensity markers can determine potential candidates for early disease diagnosis.

Despite the common belief that in transfer learning, the source data should share high-level semantic overlap with the target data, our findings (aligned by recent work by Zhao et al. (2021)) suggest that low-level features play a role. We showed that pretraining real data classification with weights of the synthetic data that has a similar low-level characteristic leads to successful information transfer. We achieved accuracies near to one for the classification of real MALDI-MS and LC-MS datasets. These results using transfer learning have been reached, while training from scratch on these datasets would diverge on some folds of data due to scarcity and high-

dimensionality.

As our biomarker detection approach was built on DNN decision interpretations, we compared the quality of different interpretation strategies, including saliency maps, modified backpropagation, and LRPs. Based on our controlled experiments on spike-in MALDI-MS datasets, we speculate better DNN explanations using LRP variants for the analysis of the noisy high-throughput proteomics data. Based on presented datasets, we observed that the performance of LRP on highlighting the spike-in regions is (a) slightly better than methods like input×gradient and input-gradient and (b) significantly better than methods like SmoothGrad, Deconvolution, and guided-backpropagation. This is mainly because LRP takes into account of the whole input, which makes it less prone to the discontinuity problem (Montavon et al., 2018) and consequently more applicable for very noisy proteomics data. The gradient methods, as the earliest DL interpretation approaches, are locally calculated, and therefore, small changes in input can cause drastic changes in the output, which is particularly not applicable for such data. In addition, unlike gradient methods that explain the whole network at once, LRP takes advantage of the structured layer of neural networks and simplifies the explanation problem. Therefore, since it decomposes the function into simpler functions and explains these easier functions, it results in more reliable explanations.

We also studied the role of the false discovery rate by adjusting the p-values in our analysis. It is demonstrated that false discovery rate can weed out the wrongly detected features. Applying this correction to real LC-MS proteomics significantly reduced the false positives. Nonetheless, the risk of losing important information should always be taken into account, as we experimentally showed that the other workflows lost the discriminating peaks by such correction.

It should also be noted that although we consider a minimal data preprocessing, these steps (Liu et al., 2020a; Kantz et al., 2019) could be potentially added to our framework for further improvement of the results. To further this work, we are interested in applying our proposed method to more real diseased cases in which the data may require some necessary preprocessing steps, such as the batch-effect correction. We consider investigating if our method can be adopted to remove this effect.

This work can be extended to the multi-subject localization of biomarkers. In this case, the interpretation of a robust multi-class classification network on the LC-MS map of samples would highlight the dominant differences of each class from the

others. These differences are the potential position of biomarkers. We also consider adopting different LRP rules to different layers of the network due to their confirmed success in machine vision applications (Samek et al., 2020).

Chapter 4

Interpretability Assessments for Enhancing DNN Classifier of High-throughput Data

As machine learning (ML) and Deep learning (DL) have been rapidly growing in real-world applications, a concern has emerged that the high-precision accuracy may not be enough in practice (Samek et al., 2020), and interpreting the decisions is important for robustness, reliability, and enhancement of a system. While in the previous chapter, we employed interpretability to realize a medical condition relevant information, this chapter demonstrates how to use interpretability to enhance the model for classifying conditions and provides robustness evidences. We quantitatively measure the quality of DNN classifier interpretation on high-throughput data for enhancing the architecture tuning. Besides, through the means of visualization we demonstrate the robustness of the classifier decisions.

The model interpretation is deemed important for deploying ML/DL models in medical decision support systems, especially when the size of good quality-labeled data is small. In such scenarios that we have also encountered during our MS analysis in the previous chapter, training a DL model requires carefully tuning the network architecture and hyperparameters. Tuning the model parameters on small sample size may encourage the network to get biased to the training instances. Therefore, the measurements based on prediction performance may not ensure generalization accuracy or introduce the optimal architecture tuning. Even with high training and generalization performance, this question may arise: what if the test data do not represent the general real-world instances. This is especially important when a model is integrated into clinical settings. One solution is to provide clinically plausible explanations, which can provide information about, for example, what are the most pertinent areas of the given data for a model to make the decision. The relevance of these areas to what we expect as a human brings more reliance on system generalization and provides users with information to decide whether to rely on the system's individual predictions.

To extend our MS analysis in this context, this chapter quantitatively assesses the classifier explanations to provide generalization evidence and enhance the neural network architecture design. We first review the background of the interpretability assessments in the literature in Section 4.1. In Section 4.2, we explain how we assess DNN classifier interpretations in high-throughput data and apply it to proteomics data. We introduce the measurements for this assessment in Section 4.3. Section 4.4 elucidates the application of interpretation assessment on tuning the architecture of a DNN classifier. We also run a sensitivity analysis to justify the robustness of explanations themselves. This analysis is highly important since the explanation

methods are criticized for being sensitive to small changes of data instances. Further, in Section 4.5, we assess the interpretations by means of visualization for scenarios where quantitative assessment is not possible due to the lack of knowledge about the underlying data. Section 4.6 elucidates why we choose DNN at first place for analyzing high-throughput data in their raw format over conventional machine learning methods through comparing the interpretation of these methods.

4.1 Interpretability Assessment and Related Works

The evaluation of explanation can be divided based on the purpose of explanation: understand the behavior of the model for the user, employ for debugging the model, or facilitate making decisions. In the following, we will give examples for these purposes.

4.1.1 Understanding the Model Behavior

Assume that to interpret the model, the explanation study introduce a set of features that are deemed important for decision making. Then, the evaluation is formulated by calculating the importance of the selected features on model performance. Qi et al. (2019) measure the importance of the selected features by removing these features from the data and observing the model's performance on the new data. Given an imaging task, they first identify important features on a single image using a saliency map, delete these features from the sample and plot the prediction probability of the model as the features are removed from the image until, for example, all the image is deleted. Then, the curve of the plot demonstrates whether the explanation model corresponds to the model's behavior or not. If the curve falls quickly, it means the explanation method picks the features that are more representative of the model's behavior. This evaluation also can be measured via inserting the important features into a flat image instead of deleting the features from the original image. In this case, raising the curve quickly in the plot would indicate the correspondence of good explanation to the model's behavior. Ghorbani and Zou (2019) use the same idea of deleting or adding important features but in the training phase. Instead of tracking the influence of features in the single sample on the prediction, the influence of

deleting or adding important features on the training phase is measured. Apart from evaluating the explanation, the data cleaning is addressed with this approach.

Another approach for evaluating the explanation according to understanding the model behavior is through simulation test proposed by Hase and Bansal (2020). In this approach, given a classifier and an explanation method of choice, the prediction on validation data and the corresponding explanation are shown to users. The users first need to figure out the model behavior based on this information. Then, new data without any labels or explanations are shown to the users, and they are asked to guess the classifier prediction on the newly presented data. Finlay, the user prediction is compared with the prediction of the actual classifier. This shows how well a user might understand the explanation strategy, which can measure if the explanation method was successful in explaining the model behavior.

4.1.2 Debugging

Here, examples of debugging a classifier through evaluating the model explanation are presented. The goal is to examine if the explanation can give an insight into a bug in a model. Ribeiro et al., 2016 address this task by first presenting a buggy classifier's prediction and true labels to the subject. They ask the subject, for instance, if they think the classier is trustworthy in the real world and how they think the algorithm is able to distinguish between classes. Then, they ask the subject the same questions after the explanation of the predictions were presented to the subject. They showed that almost all of the subjects identified the correct insight and explained a bug in the classifier after being presented with the explanations. Further, the trust in the classifier dropped substantially. This study elucidated the utility of explaining predictions in finding a buggy or a bad classifier. Alternatively, the strength of explanation for finding a bug in a classifier can be measured by comparing a clean classifier with a buggy classifier. For instance, the prediction and explanation of both classifiers are presented to the subject, and they are asked which explanation is better in explaining the reason for classifying the given sample. The more the subject picks the explanation of the clean classier, the more useful the explanation could be for detecting the bug.

4.1.3 Facilitate Decision Making

Another way of evaluating the explanation methods is to investigate if these methods can help humans make a better decision. Mac Aodha et al. (2018) proposed a teaching framework that provides feedback from a machine learning classifier on specific tasks through interpretable explanation, and determines how the human learners incorporate this information. In an image task, explanations of the classifier's prediction on different categories are shown to subjects. The subjects are asked to make the classification on unseen samples, once before and once after they are given the classifier's prediction explanation. The performances of the users would indicate the quality of explanation. Improving the users' performance after giving them the prediction explanation determines the explanations quality.

4.2 Quantitative Interpretation Assessment for Enhancement of DNN Classifier

The methods of interpretation assessment so far mainly targeted image-based tasks. This section describes how such assessments can be applied to a high-throughput data classification task that consequently helps enhancing the models and understanding its behavior. Although the strategies introduced for *understanding the model behavior* are applicable for high-throughput data analysis (which will be employed in Section 4.5), the other two categories are infeasible to use in this context. It is because the strategies proposed for *debugging* and *Facilitating decision making* are mainly dependent on the human user, and employing the human user for evaluating the interpretation of predictions on high-dimensional data seems unrealistic and time-consuming. Besides, labeling the interpretation regions could be infeasible as well. For instance, in MS analysis of unknown data, couple of thousands or millions of features need to be studied or labeled for assessment of interpretations. Therefore, in this section, we make use of synthetically generated data to systematically assess the interpretation of model predictions.

In other cases that the underlying data are known and therefore only a subset of dimensions seems to need to be studied, the interpretation still highlights enormous features that should undergo investigation. For instance, the LC-MS data introduced in 3.8.1 and 3.6.1 contain 13 discriminating peaks. However due to the noise

and peaks of different concentrations across different samples of data, interpretation map misidentified a considerable number of peaks as discriminating. These peaks are false-positive peaks mentioned in 3.8, which are outnumbered the number of true discriminating ones. Therefore, using such data for interpretability assessment is more false-positive reflection than true positives. Besides, the interpretability assessment would be criticized for the small number of instances in these datasets. Therefore, we run the interpretation assessments in this section on synthetically generated data.

We quantitatively assess the interpretation of networks for classifying high-throughput MS data. To this end, we measure the reliance of different architectures on the discriminating regions when these architectures make the same prediction performance, and when they cannot be tuned better by just observing the prediction performances. As the interpretation method in this chapter, we use the LRP explanation method introduced in Section 3.5.3.

4.3 Evaluation Metrics

Here, we introduce selected metrics to evaluate the capability of interpretation heatmap, R_i^1, on reflecting the discriminating regions of the data. We describe in 4.4.1 that the simulated data are generated in the tabular format (vectors and matrices). The data are also converted to images. Hence, regions of interest are defined by pixels when we analyse images and indices when we analyse matrices and vectors.

For our purpose of MS data classification, we expect the classifier to rely on the discriminating peaks on the data. Hence, the metrics should represent the percentage of true-positive (TP) and false-positive (FP) peaks in which can be determined by intersection over union (IOU), precision, and recall defined in Eq (4.1):

$$\begin{aligned} \text{IOU} &= \frac{\text{relevant peaks} \cap \text{selected peaks}}{\text{relevant peaks} \cup \text{selected peaks}} \\ \text{Precision} &= \frac{\text{relevant peaks} \cap \text{selected peaks}}{\text{selected peaks}} \\ \text{Recall} &= \frac{\text{relevant peaks} \cap \text{selected peaks}}{\text{relevant peaks}}, \end{aligned} \tag{4.1}$$

where the relevant peaks and selected peaks are ground-truth and highlighted peaks by the interpretation. To obtain the ground truth on synthetic data, the mean of the images in the diseased group is subtracted from the mean of images in the

healthy group, and the absolute value of the result is taken. The result contains all relevant peaks and is referred to as ground-truth image (GTI). GTI is identical to the result of alternatively simulating several replicates of the extra peptide of diseased samples (using OpenMs, and TOPPAS) and taking the mean of the replications. We apply a threshold, γ_{gt}, to the GTI to ignore small perturbations generated by LC-MS quantification error. The spatial location of peaks is distributed widely, and therefore, we restrict our attention to the peaks with the highest intensities and set to zero a box window with a size of $[w, h]$. To this end, first, the index of the highest intensity value on GTI is selected. Second, the surrounding peaks in the window of w and h are set to zero. Next, we iterate this process until all the high-intensity regions are covered. We refer to the resulting as ground truth peak map (GTPM). The selected peaks in Eq 4.1 are extracted similar to GTPM from the LRP relevances and form prediction peak map (PPM). The metrics of Eq 4.1 can be rewritten as follows:

$$\text{IOU} = {}^{2(\sum_{(x,y)\in\mathcal{I}} \text{GTPM}(x,y).\text{PPM}(x,y))}\big/_{\sum_{(x,y)\in\mathcal{I}}(\text{GTPM}(x,y)+\text{PPM}(x,y))} \tag{4.2}$$

$$\text{Precision} = \sum_{(x,y)\in\mathcal{I}} \text{GTPM}(x,y).\text{PPM}(x,y)\big/_{\sum_{(x,y)\in\mathcal{I}} \text{PPM}(x,y)} \tag{4.3}$$

$$\text{Recall} = \sum_{(x,y)\in\mathcal{I}} \text{GTPM}(x,y).\text{PPM}(x,y)\big/_{\sum_{(x,y)\in\mathcal{I}} \text{GTPM}(x,y)}, \tag{4.4}$$

where $\mathcal{I}$ covers the entire range of (m/z,RT) values.

4.4 Experimental Design and Results

4.4.1 Data Simulation

Experiments in this section target the enhancement of the system through interpretations. To guarantee control results, we run the experiments in this section on synthetically generated MS data. As a quick recap to its introduction (Section 2.1.4), LC-MS consists of two levels of separations; First, a protein solute (mobile phase) passes through a chromatography column (stationary phase), which effectively separates the components based on the chemical affinity and weight. RT measures the time taken from the injection of the solvent to the detection of the components. Second, each component is ionized and scanned through a mass spectrometer that generates a mass spectrum (MS). Each MS scan measures m/z values of charged particles and peak intensities. Stacking all MS scans on top of each other forms

three-dimensional data whose x, y, and z axes are m/z values, RT, and ion-count intensities, respectively.

To generate the synthetic LC-MS dataset, two groups of samples representing healthy and diseased classes are simulated using UniPort human proteome dataset (Consortium, 2019). The healthy class contains 20 peptides. Two peptides that are independent of the peptides in the healthy samples are added to the peptides in the healthy group to form the diseased group. As a result, there are 20 and 22 peptides in the healthy and diseased group. The accession number of the corresponding sequences is provided in Table 4.1. The quantification of two extra peptides in form

Table 4.1: The accession number associated with diseased and healthy group in the synthetic dataset.

Classes	Peptide sequences
Healthy	Q9NYW0, Q9NYV9, P59538, P59539, Q96CE8, Q96A56, O75478, Q86TJ2, Q15543, Q15573, Q9H5J8, O00268, Q9UI15, Q9H2K8, Q17R31, P10636, P68366, A6NHL2, Q13509, Q9NVG8
Diseased	Q9NYW0, Q9NYV9, P59538, P59539, Q96CE8, Q96A56, O75478, Q86TJ2, Q15543, Q15573, Q9H5J8, O00268, Q9UI15, Q9H2K8, Q17R31, P10636, P68366, A6NHL2, Q13509, Q9NVG8, Q9HA65, Q9ULP9

of peaks on LC-MS map of the diseased group define the discriminating peaks on which we want the classifier to rely the most for classification decision. Investigating such differences not only evidences of a reliable classifier as it is the purpose of this section but also is considered as the basis of diagnosis of different biological conditions (discussed in Chapter 3) and disease treatment —, e.g., measuring the concentration level of cardiac troponin that enters the blood soon after a heart attack, or measuring thyroglobulin, a protein made by cells in the thyroid, which is used as a tumor marker test to help guide thyroid cancer treatment.

The data then is read and converted to mzML by OpenMS MSSimulator (Röst et al., 2016), with the following settings: label-free quantification, Trypsin for digestion, and electrospray ionization (ESI), random noise selected between biological and technical noise at each run, m/z range of 2000, retention time range 240 min, minPeaks = 2. The rest of the parameters remain the same as the default. The samples are then read by TOPPAS (Kohlbacher et al., 2007) to generate images. The width, height, and pixel intensities of images present m/z, RT, and ion-count

intensity, respectively. It should be noted that the images still represent the raw data. The only difference between the matrix of raw data and the converted images is that the ion-count intensity range in raw data is scaled to [0,255]. We also run the pipeline and all the experiments on the raw data matrix to make sure the scaling would not cause loosing small peaks in our analysis. No noise filtering, spectra filtering, or other corrections are applied in this stage. The only noise reduction we applied to the raw data is that we remove the ion-count intensities less than two of the data. The dataset contains 4000 samples of each group. 10% of each group is left out for testing, and the rest is used for training and validation.

4.4.2 Optimizing the Model Parameters on Synthetic Data

In this section, we use interpretability assessment to tune CNN model parameters. Specifically, we show how to properly tune the parameters of the network when changing some layers does not considerably change the prediction performance.

Many standard well-structured networks such as Vgg18, ResNet32, and InceptionV3 have been established a great performance on huge datasets of millions of instances, such as ImageNet and they have been applied successfully to many applications in the medical domain as well. However, as it is explained in Chapter 3 these standard networks such as ResNet32 fail to fit on high-throughput proteomics data. This can be explained by the fact that MS and LC-MS data contains more local dependent features than global ones. On the other hand, a large receptive field in very deep networks encourages the network to capture global dependencies as well. Hence, network may learn some global pattern, for instance, patterns related to the global noise. To avoid this problem using standard deep networks, we design and tune the network's architecture from scratch. We tune the number of layers, the number and the size of filters, hyperparameters, etc, using SGD optimization and tracking the accuracy and loss of training and testing samples. But, our experiments show that changing some layers on the network architecture does not considerably change the accuracy and loss of training and testing phases. In such cases, either more layers with more learnable parameters would be chosen to obtain a better generalization performance, or a shallower network is employed in favor of the need for less memory, for example, for the development in mobile system applications. The purpose of our experimental study in this section is to demonstrate how adding or removing the layers mentioned above could make a difference in the focus of the

network on the relevant information of the data for decision making by assuming fix number of training instances.

Network Architecture Selection

We study the influence of varying one to two fully connected layers (FCL), convolutional layers (CL), and max-pooling layers (MPL). Although the variation of these settings results in slight differences in the classification performances, our experiment highlights the major improvement in their interpretations. We elucidate this improvement through assessing the classification decision reliance on true discriminating regions of the data.

Due to the shortage of annotated real datasets at the discriminating peaks level, the proposed model is developed and tuned on a synthetically generated dataset. Another important reason to run the experiment on synthetic data as it is described previously is to guarantee control results of the effect of network architecture on its interpretation and, consequently, its generalization.

To demonstrate this impact, we tune parameters of the network, including the number of FCL, CL, and MPL for classifying synthetic LC-MS data. These parameters do not change the classification accuracy in the variations presented in Table 4.2; however, they significantly had an impact on the focus of the network on the relevant part of the data for making decisions. This effect is measured through interpretation assessment metrics in this table. To form this table, networks that are built by varying parameters mentioned above are first trained and interpreted using the LRP interpretation strategy. Then the interpretations are assessed through IOU, Precision, and Recall. This evaluation is presented in Table 4.2.

In our assumption, networks with higher values of IOU, precision, and recall are more generalized due to the fact that these networks know on which part of the data look for the reason of distinguishing a sample in one class from others; therefore, it is more likely to do the same on unseen data and more likely to gain trust through the reasoning. According to the research in the DL field, exploiting deeper networks is recommended for better generalization as they offer richer representation. Contrary to the expectation, results in table 4.2 show that the interpretation of deeper networks (more CL and FCL layers) for LC-MS classification show less reliance on the discriminating peaks. As a result, among the networks with the same accuracy performance, the one with four CL, one FCL, and one MPL reach the best inter-

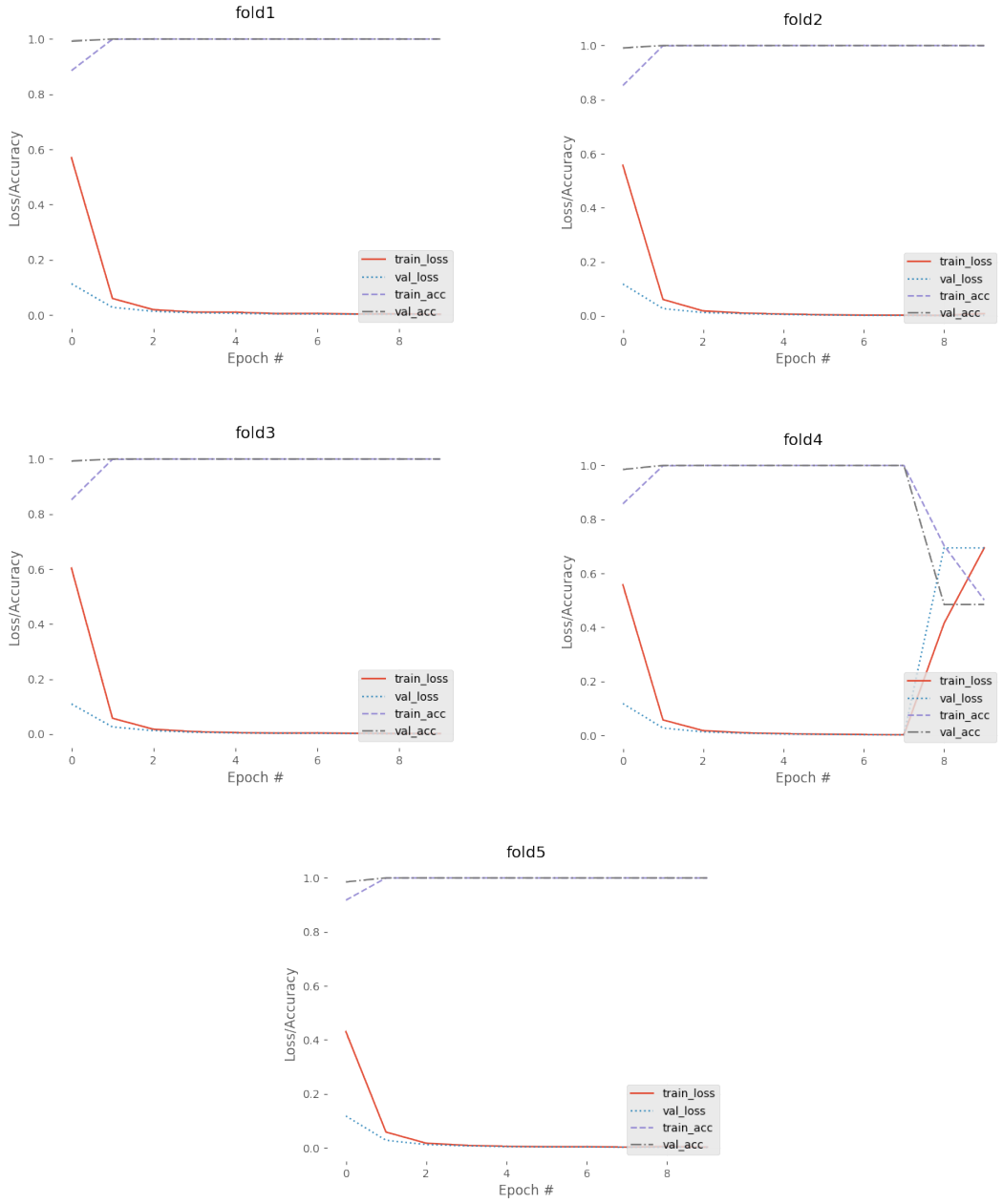


Figure 4.1: Training and validation performance of the Enhanced Classifier on LC-MS synthetic dataset. Training and validation classification accuracy are shown in the purple dash line and back dash-doted line, respectively. Training and validation losses are also shown along with the accuracies in the red line and blue dotted line. This plot demonstrates the five-fold cross-validation training curves for ten epochs. However, for the classification comparison, interpretation, and feature selection, the early stopping has been considered. Therefore, training is stopped after five epochs on the fourth fold, which avoids divergence.

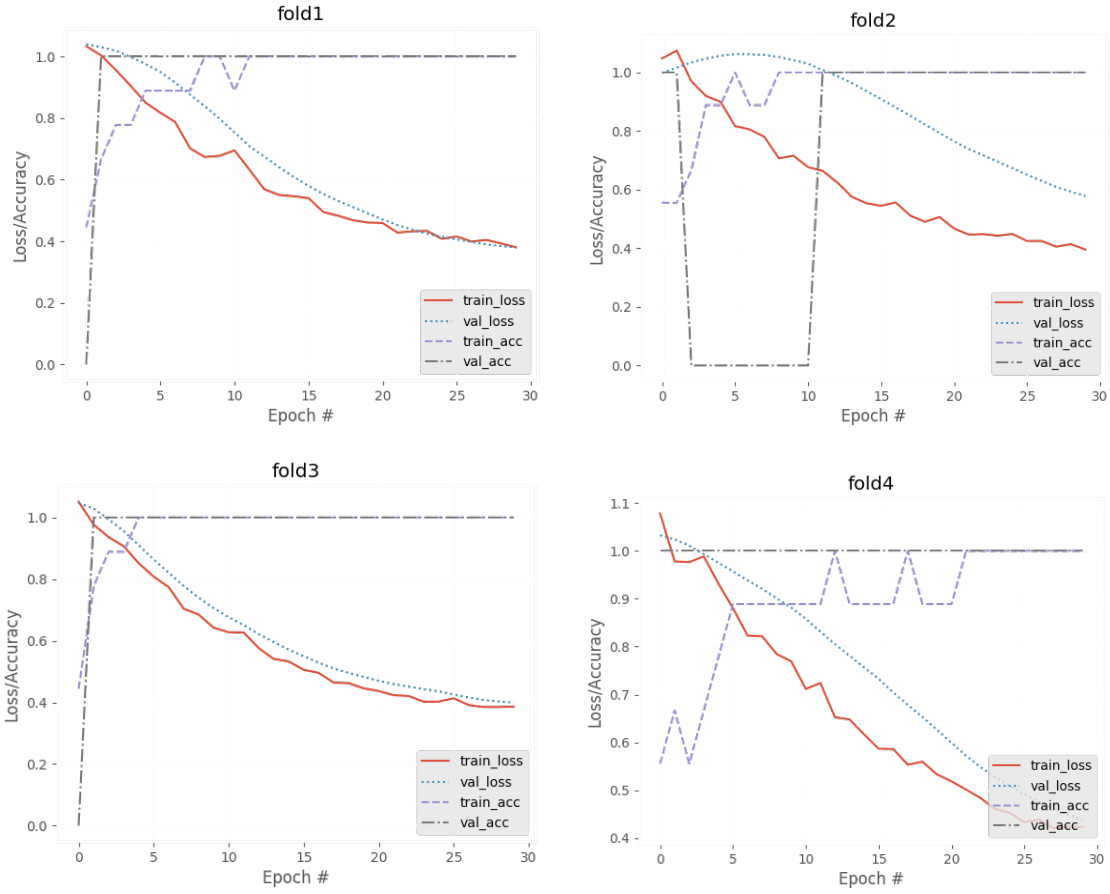
Table 4.2: Network architecture selection through interpretation assessment. This table shows the effect of adding fully connected layers (FCLs), convolutional layers (CLs), max-pooling layers (MPLs) on focusing the network on the discriminating peaks for decision making. The parameters are tuned according to the intersection over union (IOU), precision, and recall. The effect of incorporating the interpretation of diseased samples' mean (R_d) and the interpretation of healthy samples' mean (R_h) on peak detection is also demonstrated.

# CL	# MPL	#FCL	Samples	IOU	Precision	Recall
6	4	2	R_d	0.3975	0.3814	0.4149
6	4	1	R_d	0.5006	0.4513	0.5621
6	4	1	R_d, R_h	0.6177	0.6188	0.616
4	3	1	R_d	0.6599	0.5985	0.7353
4	3	1	R_d, R_h	0.7008	0.6756	0.7281
4	1	1	R_d	0.7165	0.6171	0.8441
4	1	1	R_d, R_h	**0.8501**	**0.8554**	**0.8448**

pretation performance. Hence, we can make sure that by making decisions on the health status of LC-MS samples, the network distinguishes the samples according to the data regions that are truly discriminating. This is the way how we as a human would make decisions. Therefore, we can hope for more generalization performance for the real-world instances whose perturbations are not predictable.

The classification performances of the designed network on simulated and real LC-MS data are depicted in Figures 4.1 and 4.2, respectively. The plots demonstrate that the network designed for synthetic data works as expected on the real data. It is worth mentioning that our LC-MS biomarker detection approach in Chapter 3 was built on top of this tuned model. To further validate the model selection on the real data, it is interesting to observe that the suboptimal architecture in Table 4.2 leads to worse biomarker detection performances on the real dataset, as well. However, this experiment could only validate the model selection if the biomarker detection results were only dependent on the architecture tuning. However, not only the architecture tuning but also pretraining the weights affected the outcomes of real data biomarker detection. Note that transfer learning was used to transfer the knowledge from synthetic data classification to real data classification. Thus, since the suboptimal architectures were unable to be fitted to simulated data (i.e., the interpretations result in poor IOU, Precision, and Recall) the weights can not be technically used as source weights for the real data classification. For instance, our experiments showed that a suboptimal architecture (e.g., L=6, MPL=4, FCL=2) in Table 4.2 barely can

find the true biomarkers on real data. Although this observation is aligned by what we have expected, it cannot validate the model selection. It is because to have a fair comparison with the performance of the selected architecture, we should pretrain this network using the simulated data weight, as well. But, the results in Table 4.2 show that the suboptimal architecture itself poorly highlight the discriminating regions of the simulated data. Hence, the poor biomarker detection performance on real data either can be caused by the selected architecture or it can be caused by the poor initialization.



See the next page for a detailed description.

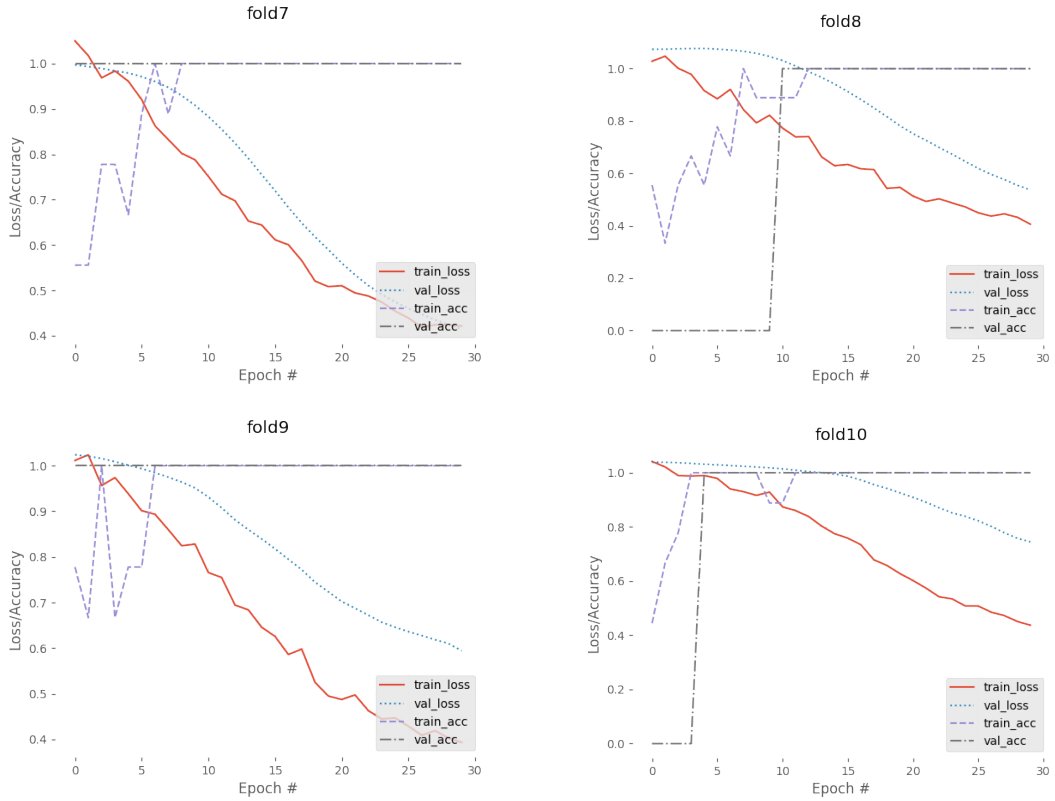


Figure 4.2: Training and validation performance of the Enhanced Classifier on LC-MS real dataset. We retrain the network that has been initially trained on the synthetic LC-MS data (shown on Figure 4.1) on the real LC-MS data. Training and validation losses are also shown along with the accuracies in the red line and blue dotted line. Training and validation accuracies are shown in dash line and back dash-doted line, respectively. It can be seen from the plots that all folds of data has reached the maximum accuracies.The trends are less smooth than simulated data because of the smaller data points in the real dataset than the simulated dataset.

Interpretation Importance Across Different Classes

So far, all the assessments are formed based on the information gained by interpretation of disease instances in the data. Now, we would like to make an extra point about the use of both class interpretations for the analysis of LC-MS data that can alleviate the problem of false-positive detection. To this end, we change the gear a little to our goal of biomarker detection approach in Chapter 3. In this section, we experimentally explain the effect of incorporating the interpretation of healthy predictions along with the interpretation of diseased predictions on biomarker detection performance. In Section 4.3, we have described in detail how prediction peak

map (PPM) is calculated through LRP relevance values. As a recap, To estimate relevance values on the training set, we calculate the mean of the diseased samples, run the trained network on the mean, and calculate the relevances. By convention, positive relevance values are the evidence of existing relevant peaks belonging to the respected class. Therefore, in our study, positive relevance values on the interpretation of diseased class are associated with the biomarker candidates. Basing these candidates for further biomarker analysis – that described in detail in Chapter 3 - keep too many false-positive peaks into consideration that can be decreased by incorporating the interpretation of healthy samples along with the interpretation of diseased samples. The positive relevances of the interpretation of the healthy group can be explained as the absence of diseased relevant peaks or the presence of healthy relevant peaks. Because all the peaks in healthy samples are presented in diseased samples in our simulation, the positive relevances of this group are explained as the absence of diseased relevant peaks. Accordingly, the indices of high-ranked relevances on the diseased group are selected as biomarker candidates if the corresponding indices in the interpretation of the healthy group attribute non-negative relevance. The results cause by this extra criteria are shown in Table 4.2. As it is apparent, IOU and Precision, which are both directly affected by FP in the denominator, have considerably improved. We also confirm the effectiveness of this idea in real LC-MS data as well in Table 4.3.

The improvement in the false positive rate on the LC-MS real data analysis was not as pronounced as the LC-MS synthetic data analysis. This behavior can be statistically explained by the number of samples in the synthetic dataset (~ 8000) that outnumber the real dataset (~ 10). We calculated the interpretation analysis on the mean of the samples' intensities. Therefore, the mean intensities on the large set of data represent whole data distribution better than a small set. Consequently, the importance of features belonging to the larger dataset assigned by the network's decision would be more precise.

4.4.3 Interpretation Sensitivity Analysis

The sensitivity analysis of deep learning interpretation methods has recently gained attention intending to address how much we can trust in the outcome of interpretations. For example, Arun et al. (2020) discussed that it is essential to rigorously examine the utility and robustness of the explanation in the context of medical ima-

Table 4.3: The effect of incorporating the interpretation of diseased samples (R_d^1) and the interpretation of healthy samples (R_h^1) on peak detection.The third and fourth rows demonstrate the number of features satisfying two representative criteria, including t-test with multiple hypothesis testing (q-value$<$ 0.05) and fold change (FC $>$ 10). The plus sign denotes the combination of different criteria. The numbers written in parentheses indicate the selected biomarker peaks. The results show that incorporating both existing classes' interpretations contributes to decreasing the false discovery rate.

	R_d^1	R_d^1,R_h^1
# All selected features	8044 (12)	6992(11)
t-test ($q < 0.05$)	3985 (11)	3499(11)
t-test ($q < 0.05$) + FC (> 10)	222 (9)	195(9)

ging data. This study posits that explanation trustworthiness requires repeatability, reproducibility, and other imaging data analysis assumptions. In addition to repeatability and reproducibility in the context of MS feature importance discovery, we posit that the explanations need to be consistent from one sample to other samples of the same group in order to guarantee the robustness of the results. We assess this assumption by comparing the intersection over union (IOU) when the network is run in cross-validation mode. The visualization scheme of this examination is depicted in Figure 4.3. This experiment specifically run on the synthetic data in order to avoid problems of disentangling errors made by the model from errors made by the explanation. First, 10% of the data is left out for testing, and the rest is used for training and validation sets in a five-fold cross-validation split. On every cross-validation run, the network is trained on the training set. Then, the inference is run for testing and validation set, and finally, LRP interpretation is run on the predictions. The interpretations of the test set, which are generated five times over five-fold cross-validation, reach almost 99% IOU. The high level of overlapped regions demonstrates the reproducibility and repeatability of the interpretations. Likewise, the interpretations of the five validation sets over the training using five-fold cross-validation reach almost 98% IOU. This result shows the robustness of the interpretations with respect to changing the samples in the data.

These results not only justify the stability of the interpretations and the designed classification but also imply the robustness of feature selection results in Chapter 3.

4.5 Visualization Assessment of Interpretations

So far in this chapter, we have shown the methods of assessing the models' interpretation, where the training data's discriminating features are known as the ground truth, or this information is simulated and assessed on the synthetically generated data. This may raise concerns about what if this level of ground truth cannot be obtained or be simulated, which makes it impossible to evaluate the interpretations through the IOU, precision, and recall (as it is carried out in Table 4.2). Since, in this case, we do not know where to expect the model to rely on to make decisions, the interpretation assessment cannot be quantified. We instead employ the following evaluations for MS data, similar to visualization methods introduced earlier in Section 4.1:

1. Feature importance: Measure the importance of highlighted areas from the output of the interpretations for the model's performance. To this end, we add the high-ranked features that are achieved by the interpretation of the model to the data one after the other and plot the model's performance. The steeper the plot's slope, the more important the selected features are for the model, which determines the quality of the interpretation method.

2. Feature visualization: Visualize the areas that the network thinks are the most pertinent through its interpretation and check if these areas are logical and are coherent to us as humans by what the model is trained for.

4.5.1 Measure Importance of Features

Figure 4.4 shows the interpretation assessment through feature importance described in item 1. This evaluation is compared to linear SVM with ℓ_1 norm (ℓ_1-SVM) (Noble, 2006; Haq et al., 2019), as a conventional feature selection models for sparse solutions. The experiments run on MLDI-MS datasets, spiked160, spiked80, and Pancreas cancer datasets, introduced in Section 3.6.1 and 3.6.2. To form this plot, first, the important features that are selected according to the feature selection method proposed in Chapter 3 are ranked based on the relevance values that LRP assigns to the features. These features are inserted one by one to a vector for each sample, and the model each time is separately trained on these data. The accuracies are plotted

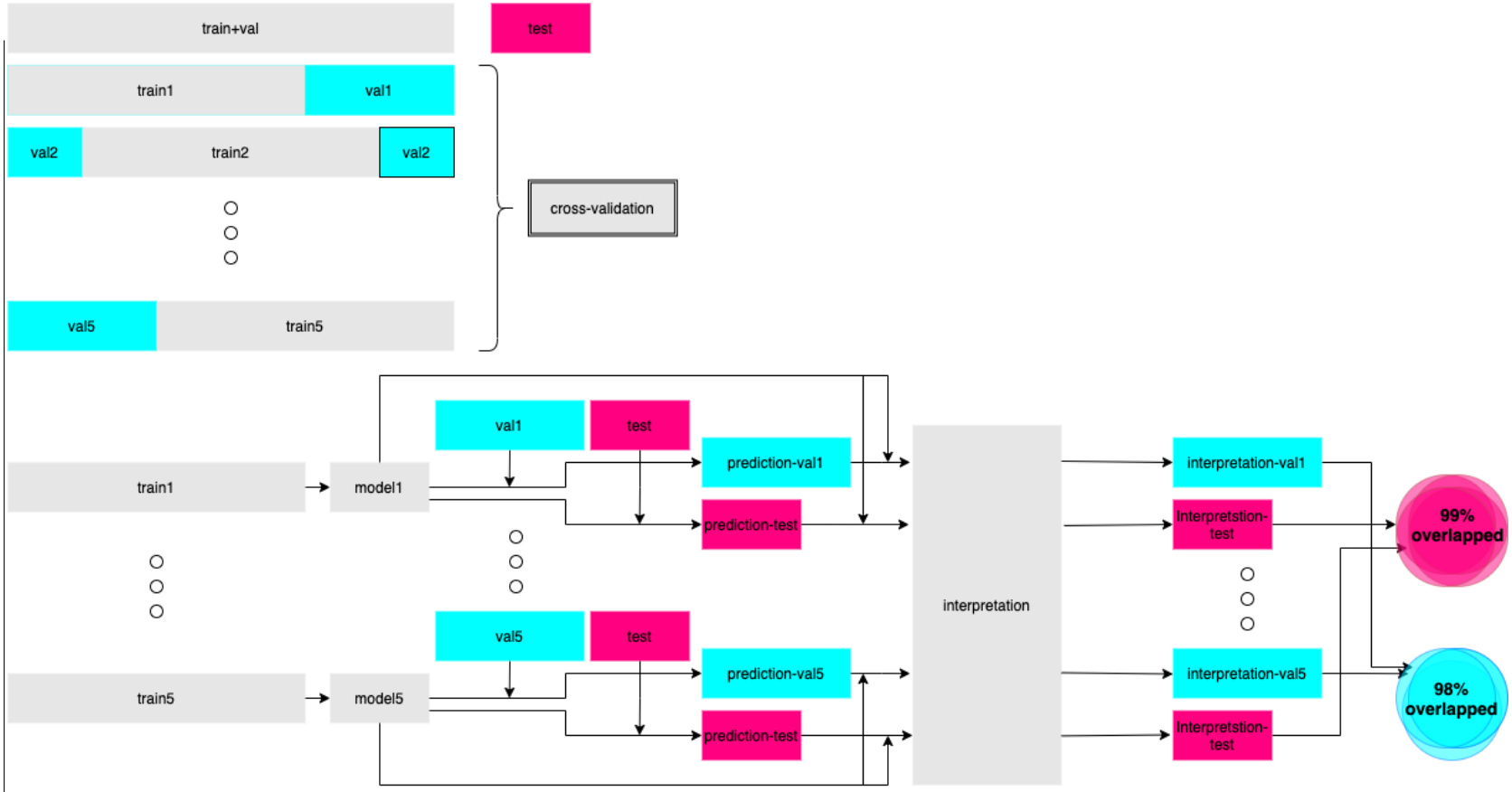


Figure 4.3: LRP Sensitivity analysis illustration through cross-validation. The block on the top shows how the training, validation, and test set are distributed. The bottom diagram demonstrates how these sets are used to measure the repeatability and reproducibility of the prediction interpretation. Running the validation and corresponding interpretation on a different fold of data and the high overlap score of the results justify the robustness of the interpretations with respect to changing the samples in the data. A high overlap score of the prediction interpretation of these networks (that has been trained on the different fold of data) on the test set demonstrates the reproducibility and repeatability of the interpretations.

in Figure 4.4 with respect to the number of features that were added to classify the dataset. It can be seen that our model reaches the maximum performance on all three datasets with the first few high-ranked features. It is worth noting that these features have been found without any prior knowledge of the discriminating regions on the whole data. The sharp steep of the curve demonstrates the high quality of DNN interpretation.

Compared with the plot by SVM interpretation, the plot's slop by DNN interpretation is sharper. This means the features that are extracted according to the DNN interpretation approach reach the maximum classification performance faster and with fewer features. This is an important property in situations where selecting more features leads to higher costs in some biomedical pipelines in which each feature must be validated in expansive wet-lab experiments.

4.5.2 Visualization of Important Features

Figure 4.7 shows the interpretation assessment through feature visualization described in item 2. These plots show the interpretation of the first 12th regions that DNN relies on the most to make classification decisions on real LC-MS data. As mentioned earlier, in this experiment, we assume that the discriminating regions are unknown. We classify the samples into healthy and diseased groups, which LRP then interprets. The feature selection approach proposed in Chapter 3 is applied, and the first 12th high ranked regions of the data are visualized. The samples classified as diseased are visualized together at the location of high-ranked regions, indicated by MP. Likewise, The samples classified as healthy are visualized together at the same locations, indicated by noMP. By looking at these plots, it is clear that some peaks in one group are missing in the other group. If we assume that the model function is unknown, but we know that the model makes decisions based on these regions, we can easily speculate that the model does a classification task. Therefore, it can be concluded that the interpretation method is meaningful and coherent by the classification task's aim. In our study, we have already justified the interpretation method's quality in different ways. But, if we were limited to assessing the interpretation only by the feature visualization method, the non-expert feedbacks on the feature visualization would be necessary to demonstrate the effectiveness of the interpretation method.

With these two experiments, we have shown that, given a dataset of two groups

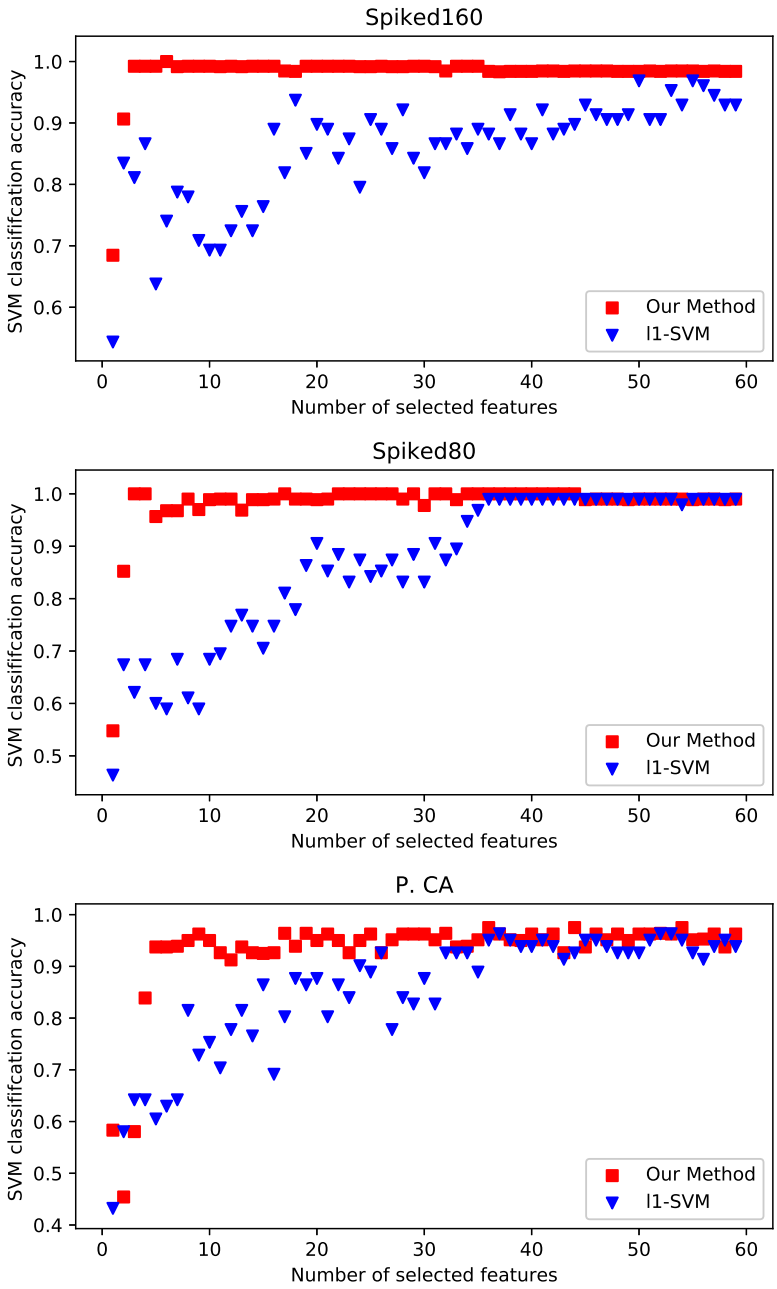
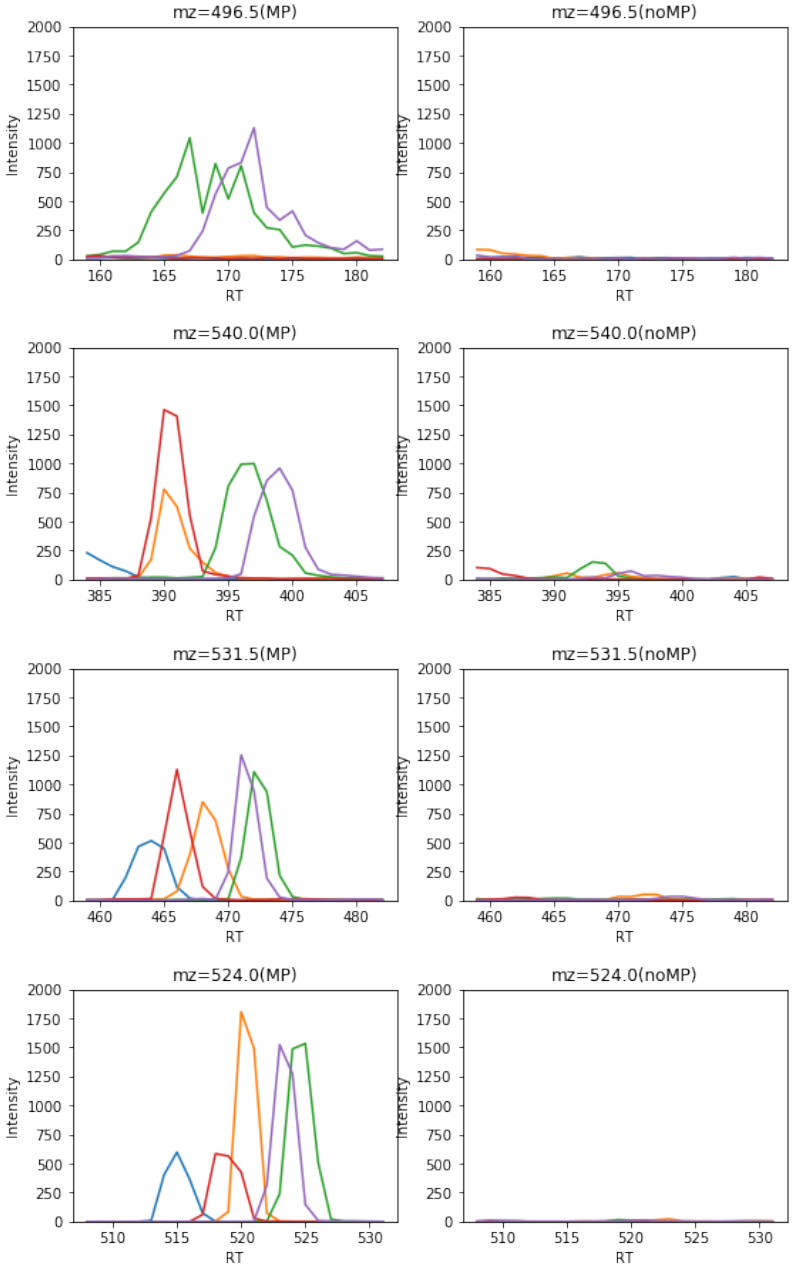
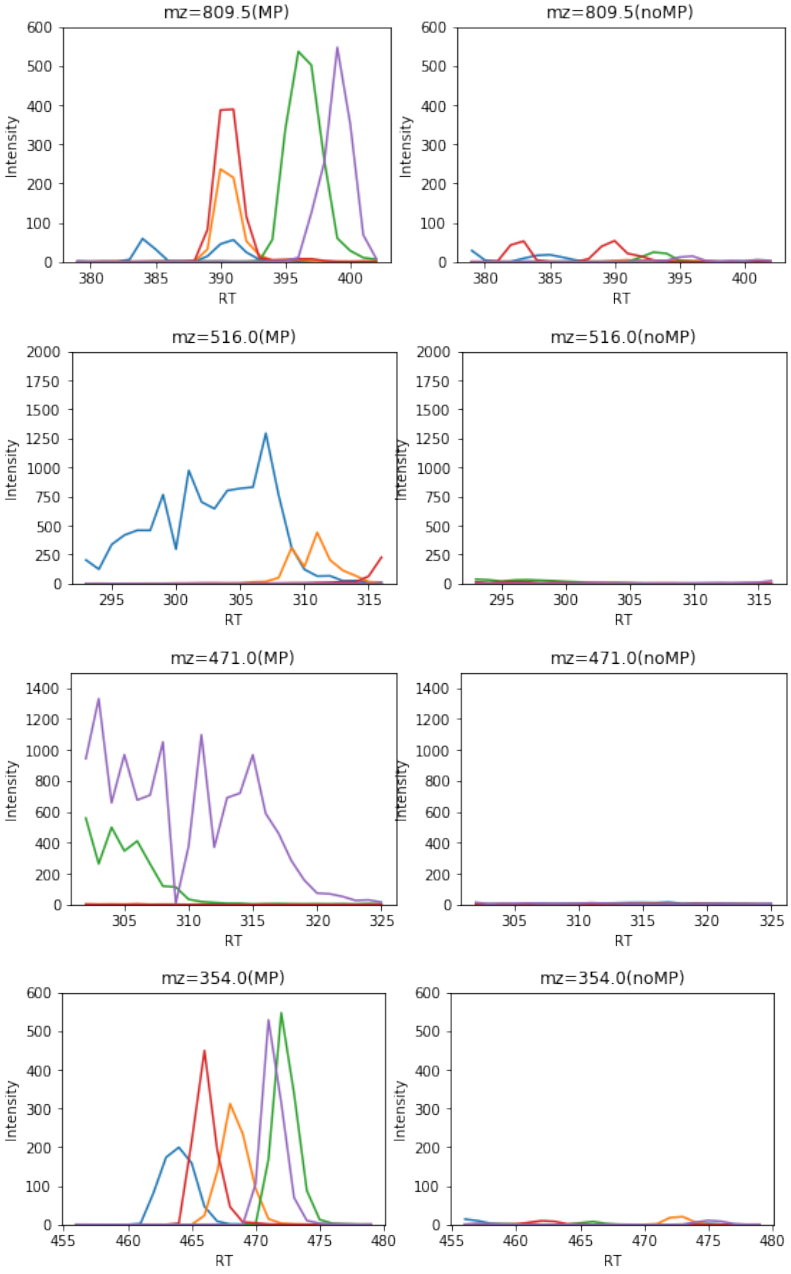


Figure 4.4: Interpretation assessment through visualizing the model performance while adding the selected important features to the data. Plots show the strength of selected features on spiked160 (first row), spiked80 (second row), and P. CA (third row) using our method in red-square and ℓ_1-SVM in blue-triangle.

without any prior knowledge about discriminating regions of the instances, the quality of the interpretation of the DNN classifier can still be assessed. Besides, the interpretation output may also lead to learning the underlying characteristics of the data. For example, suppose the cause of a specific disease or effectiveness of a drug is unknown in the protein blood samples. In that case, this visualization can illustrate which features of the data are most affected by the sickness or a specific drug.



See the next pages for the remaining plots and a detailed description.



See the next pages for the remaining plots and a detailed description.

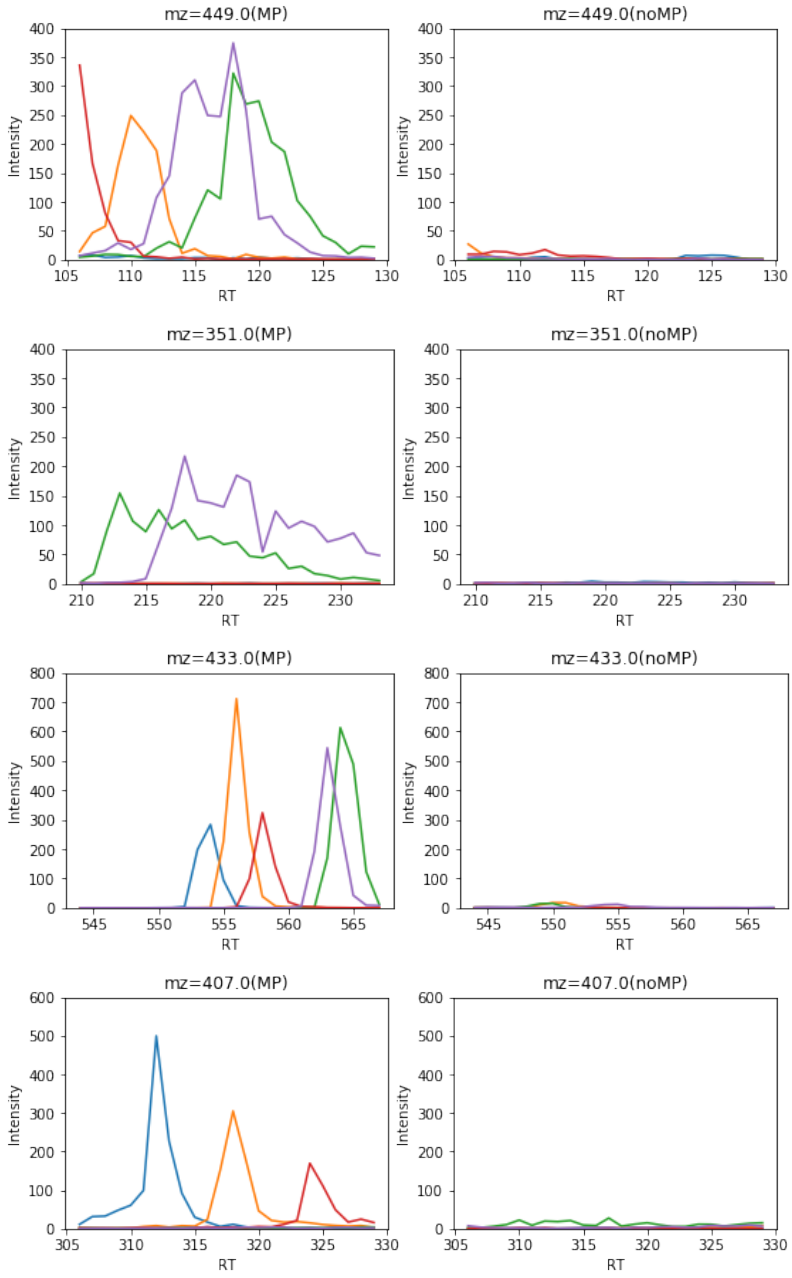


Figure 4.7: Visualization of the first 12 regions on which the network relies to make classification decisions for the LC-MS real data. If this visualization is given to a user without any prior knowledge about the network functionality, the user are likely able to describe the network as a discriminator or a classifier.

4.6 Interpretation of Conventional Machine Learning Classification Models

In the course of experiments in this thesis, all the high-throughput data analyses are carried out directly on the raw instances. We avoided any dimension reduction before data analysis because we wanted to keep all the dimensions and hinder losing important information. However, keeping all the dimensions when analyzing high-dimensional data with ML algorithms often leads to overfitting. It means a model that works well on training instances likely performs poorly on the test set. This can go to even the worst scenarios, where the model performs well on the test set too, but after deployment, the performance degrades on new instances. That is because you cannot ensure if the test data represents all unseen instances and deformations in the world. In such scenarios, the model's interpretation can play an important role. If the interpretation highlights relevant regions of the data, for example, discriminating region in a classification task, it means the model does not make decisions based on irrelevant artifacts of training instances. Therefore, it is more probable to be robust against the new samples' artifacts. But, the problem is that it is not always possible to interpret ML models that are fitted on high-dimensional data.

To exemplify this shortcoming, we run experiments for classifying LC-MS data (4.4.1 and 3.8.1) as one of the high-dimensional datasets investigated in this thesis using conventional ML algorithms and compare the performances with DNN. Table 4.4 shows the classification comparison of ML methods, including support vector machine (SVM) with linear kernel, decision tree (DT), and Adaboost with our CNN model. The parameters of the selected methods are tuned using grid search in Scikit-learn on synthetic data. We run the CNN architecture that we tuned in Section 4.4.2. We use five-fold and leave-on-out cross-validation for training on the synthetic and real datasets, respectively. As it is apparent from Table 4.4, there is a huge gap in the classification performance of ML methods between the synthetic data and the real data. As has been already discussed, one way to investigate the reason is to interpret the results. One way to interpret ML models is model agnostic methods, which enables estimating the importance of features for decision-making by any trained model regardless of the model's complexity. For instance, permutation feature importance, measured by randomly shuffling the feature and tracking the drop in the model's score. LIME (Ribeiro et al., 2016) as another model agnostic inter-

Table 4.4: Classification comparison of the convolutional neural network (CNN) with conventional machine learning methods including, decision tree (DT), support vector machine (SVM), and adaboost. CNN shows significantly better classification performance on the real datasets. The interpretation is not available for weak classifiers. On the synthetic dataset, ML methods are as accurate as CNN. However, SVM interpretation demonstrates the overfitting effect. Interpretation of the synthetic data is reported by intersection over union (IOU) between the selected and true peaks. Interpretation of the real data is reported by the amount of true positive peaks from 13 spike-in peaks. '-' shows no interpretation is available for the models.

Synthetic dataset	Accuracy	Sensitivity	Specificity	Interpretation (IOU)
SVM	0.98	0.99	0.98	feature importance(< 0.1)
DT	1.0	1.0	1.0	-
Adaboost	0.99	1.0	0.99	-
CNN	1.0	1.0	1.0	LRP (0.85)
Real dataset	Accuracy	Sensitivity	Specificity	Interpretation (TP/13)
SVM	< 0.5	< 0.5	< 0.5	-
DT	< 0.5	< 0.5	< 0.5	-
Adaboost	< 0.5	< 0.5	< 0.5	-
CNN	0.8	0.8	0.8	12/13

pretation locally interprets any model around a single prediction. Given a trained model, LIME perturb each instance locally, calculates the distance of the perturbed instance from the original sample according to the trained model, and generates a new dataset. A linear model is then fitted on the new dataset. The linear model coefficients determine which features are more dominant. These methods, however, are computationally infeasible for analyzing high-dimensional LC-MS data. On the other hand, inherently interpretable models cannot correctly classify complex LC-MS data. As an example of such models, we can name linear models in which the weights of the variables serve as the explanation or shallow decision trees in which the normalized total reduction of the Gini index by every feature yields the explanation. In Table 4.4, despite adaboost that is not inherently interpretable and decision tree (DT) that is not shallow enough to be interpreted, linear SVM still can be explained by the weights assigned to the features. According to this table, SVM reaches comparable classification performance as CNN. However, the explanation results in a very poor IOU – less than 10% – between the important features selected by the coefficient of the SVM model and actual differences. This effect –the high accuracy and weak explanation – resulted by SVM can be explained by low fidelity of the model's interpretation or overfitting of the model caused by some biases or patterns

(comes with the simulation), unrelated to actual differences. But, the overfitting effect is more likely since SVM with the same parameter setting, trained on the synthetic data, results in a very poor classification of the real data. The adaboost and DT classification gap between the real and synthetic data can also explain the overfitting effect.

These experiments demonstrate one of the reasons that we choose DNN models for High-throughput data analysis, which exemplify not only the DNN enables reaching the high performance, but also their interpretation is now more alleviated by the recent interpretation technologies.

4.7 Discussion

The main goal of this chapter was to provide a comprehensive study of DNN interpretability assessment in the context of high-throughput data analysis and to show how this assessment can be employed to enhance the model and justify the system's reliability. These arguments were investigated with the application of mass-spectrometry data classification and biomarker detection. We classified an LC-MS dataset with a convolutional neural network that has been introduced in Chapter 3 and interpreted the results using the LRP interpretation method.

We assessed these interpretations in three scenarios: 1) the ground truth data is available, where the goal is to confirm that the predicted decisions were made based on relevant features, 2) the ground truth is not available, but similar data can be synthetically generated, where the goal is to enhance the model through interpretation assessment, and 3) there is no knowledge about the underlying data, where the goal is to confirm the reliability of detected patterns through feature importance and feature visualization.

In the first scenario, where the relevant information or discriminating features are known, the interpretations were assessed directly on the ground truth. This scenario was studied in Chapter 3 based on the selection of spiked-in peaks on two MALDI-MS datasets with 42381 features. We showed that most of the ground truth peaks were selected among 30 selected high-ranked features. These results justify that the predictions were made based on true features.

In the second scenario, we assume that the ground truth is not available, which is often the case for high-throughput data analysis. We proposed studying the DNN

model's behavior based on synthetically generated data. When the network is trained on the simulated data, since we know which elements we want to make decisions based on, we can check through the means of interpretation, for instance, if a certain depth of layers is able to learn the representation of such data, or adding too much pooling layer lose local dependent information in the prediction analysis, etc. Accordingly, through quantitative measurement of interpretation, we enhanced the design of the DNN so that the network focuses more on discriminating regions for making decisions. Our experiment elucidated that this information is transferable to real data analysis. We demonstrated that on LC-MS proteomics data, retraining the enhanced classifier attained maximum accuracy on all folds of 10-fold cross-validation. Note that our biomarker detection on LC-MS data in Chapter 3 was built on top of this tuned model.

In the third scenario, we assumed that not only ground-truth is not available, but also there is no knowledge about the underlying data (for the simulation purpose). We justified the reliability of learned patterns through interpretability assessment through feature importance plot and feature visualization. In the feature importance plot, we showed a steep slop by adding high-ranked dimensions, assigned by decision interpretations on three MALDI-MS datasets. The slop with DNN using our approach is significantly steeper than the conventional method using SVM. This means our approach detects the discriminating features faster with fewer false positives. This is deemed an important property for pipelines, where each feature must be validated in expansive wet-lab experiments. In feature visualization, we demonstrated the interpretations are logical and are coherent to our cognition as humans, and the model function can be estimated by only looking into the instances' interpretations.

To show the explanation's trustworthiness, we showed that the interpretations are repeatable, reproducible, and consistent from one sample to other samples within one class. Our model interpretations were shown to be stable when calculated on different data points within a class or when the model is trained on different runs. This claim is exemplified by running a sensitivity analysis. We showed the interpretations are 98% overlapped when calculated in cross-validation mode on validation sets and 99% on the test set. We calculated the overlaps on different folds of the validation set as well as the test set when the interpretations are obtained by the network trained on different folds of data.

We further showed in synthetic data analysis that exploiting the interpretation

of *both classes* rather than just the target class can considerably improve the FP in comparison with the setting when only the diseased class was considered. This observation stressed the importance of understanding the implications that are provided by interpretation analyzes. Leveraging this valuable information can foster more plausible network architectures, resulting in a more meaningful conclusion. Recent advances in the image processing field confirm this important fact (Bach et al., 2015; Samek et al., 2020; Kohlbrenner et al., 2020).

We also demonstrated why analyzing high throughput data using DNN was preferred over conventional ML models. According to Section 4.6, conventional ML models are failed to correctly fit on LC-MS real dataset. Despite high accuracy on the synthetic data, the poor interpretation of linear SVM on synthetic data and the huge gap between classification performance of real and synthetic data demonstrate the overfitting effect.

So far, the application of biomarker detection has been investigated through deep learning classification networks and their interpretation strategies in the context of high-throughput data analysis. To extract the information in our study, we only needed the class labels of samples and are independent of the labels at the biomarker level. Based on what we have learned from analyzing structured data, in the next chapter, we will continue our study in imaging modality with other architectures, such as encoder-decoder models that require being trained in a fully supervised manner.

Chapter 5

A New Deep Learning Localization-Identification Approach for High-dimensional Medical Images

In this chapter, we change our focus from high-dimensional structured proteomics data analysis to analysis of high-dimensional imaging data that is deemed essential in the healthcare diagnostic process. We propose a robust deep learning (DL) imaging pipeline for the localization of regions of interest in medical images. We target the challenging task of spinal vertebrae localization and identification, which will be addressed using supervised DL approaches.

In Chapters 3 and 4, we employed the explanation of the network predictions to discover unknown patterns and localize biological relevance features from high-throughput structured data. Our approach is independent of the costly annotations at the feature level. In structured proteomics data, we took into account the deviations caused by noise and data acquisition. Considering such deviations, we showed that the interpretations of decisions of a robustly trained model can reveal relevant features. In biomedical images, however, the deviations of the region of interest can be largely varied from one sample to another. For instance, in our application, the localization-identification of spinal vertebrae, the shape and number of vertebrae can be totally varied from one scan to another. Besides, different fields of view and many other deviations related to the imaging acquisition induce more variations. Some preprocessing steps, such as image registration, can solve the spatial variations of anatomy across different images. However, to avoid preprocessing steps that add more complexity, we employ deep convolutional neural networks (CNNs) in a supervised manner, which is capable of handling spatial variations with minimal preprocessing (Anwar et al., 2018; Zhao et al., 2021) on raw data.

To set the stage, we first give an introduction to medical imaging analysis in Section 5.1, and the difficulties of spinal vertebrae detection task in Section 5.2. Section 5.3 surveys previous related works. We formulate detection of the spinal vertebra through different machine learning concepts, including segmentation using the encoder-decoder architecture of UNet (Ronneberger et al., 2015) in Section 5.5, detection using YOLO architectures (Redmon et al., 2016) in Section 5.6, and our regression-based approach in Section 5.7. These methods are built based on CNNs as the main feature extraction part. The comparison performances of localization-identification of lumbar vertebrae is reported in Section 5.8 on heterogeneous data. We demonstrate that for heterogeneous and complex data, our regression-based CNN, leads to a more robust performance. The architectures mentioned above have achieved state-of-the-art performance in many medical imaging analysis tasks. However, they are still facing challenges when the data is heterogeneous, and at the same

time, the annotated samples are scarce. To deal with these challenges, we equip all the models with the proper data augmentation, and data enhancement using the human-in-the-loop process. Moreover, we utilize strategies that we have learned from analyzing structured proteomics data, such as network generalization through transfer learning. Our lumbar vertebrae localization-identification requires assumptions about the number of vertebrae expected to be found in an image. We further extend our proposed method to the whole spine to alleviate this limiting assumption in Section 5.9. We propose to simplify the vertebrae localization-identification task by classifying the images to different (fields of view) FOVs and then address the localization-identification of each region separately.

5.1 Medical Imaging Analysis

Medical imaging is able to look into the human body and has long been established as an essential tool in the healthcare diagnostic process. To leverage the full potential of medical images, image processing steps are necessary. In the case of high-dimensional imaging data, the healthcare experts spend an increasing amount of time manually reviewing and preparing medical images to input into different kinds of IT systems. This process is time-consuming. Besides, the results may not be consistent when they are made in different attempts or by different interpreters. The attempt of automatic analysis of medical images to assist physicians emerged since 1960's (Chen et al., 2016; KIMME et al., 1977; Semmlow et al., 1980). Systematic use of machine learning has also emerged since 1980, for instance, as a second opinion to assist radiologists in interpreting images. Conventional machine learning extracts hand-crafted features from the data that are then used as an input variable to a predictor for different tasks. The quality of conventional machine learning models, therefore, depends on the domain expertise and the capability of the mathematical formulations or empirical image analysis techniques designed to translate the image characteristics to numerical values. Besides, the finite number of feature descriptors in conventional machine learning approaches may not be able to translate all the discriminating characteristics of complex data into feature space. Hence, the outcome of the model is limited to the design of the features and consequently to the domain knowledge. With the advent of powerful modern machine learning techniques such as DL, representations are learned automatically, without manually designed features.

As introduced in detail in Chapter 3, Section 3.1, CNNs learn representations using convolutional operation. When this operation is performed hierarchically in consecutive layers, it enables CNNs to learn complex pattern by transforming the data into multiple levels of abstraction. CNNs learn to extract features from training samples for a given task by iteratively updating the kernels with stochastic gradient descent. The tremendous progress of CNNs in predictive analytics almost rivals human performance in vision tasks, such as image classification (He et al., 2016a), recognition (Sun et al., 2017), or segmentation (Ronneberger et al., 2015). These successes have brought high expectations that DL, or artificial intelligence (AI), can bring revolutionary changes in healthcare and medical image diagnosis. For instance, early studies of integrating DL into computer-aided diagnosis (CAD) in radiology have shown even though deep CAD diagnosis did not reach human-level accuracy, the radiologists' accuracy was improved significantly when reading with CAD as a second opinion (Chen et al., 2013). Despite achieving state-of-the-art performance, this field is still facing challenges, in particular with the scarcity of labeled data, mostly due to the high costs of acquiring expert annotations. In this chapter, we circumvent this shortage by incorporating advanced data augmentation, transfer learning, and human-in-the-loop process. We study different supervised DL models, formulated as organ detection, on MRI images of human body. Organ detection is the backbone of numerous clinical applications, for instance, the study of anatomical structure (Fischl et al., 2002; Tu et al., 2008), diagnosis of diseases (Silveira et al., 2009; Chrástek et al., 2005), localization of pathology (Ghafoorian et al., 2017; Trebeschi et al., 2017), treatment planning (Fortunati et al., 2013), and computer-integrated surgery (Chen et al., 2016). Our analysis focuses on spinal vertebra localization-identification on MRI images. This task remains challenging mainly due to the similar appearance of neighboring vertebrae, spine deformation, different and limited FOVs, and image artifacts induced by surgical implants. Through a comprehensive study of CNNs in this chapter, we propose a robust pipeline to tackle these challenges for localization-identification of lumbar vertebrae and finally extend our pipeline to whole spinal.

5.2 Lumbar Vertebrae Localization-Identification

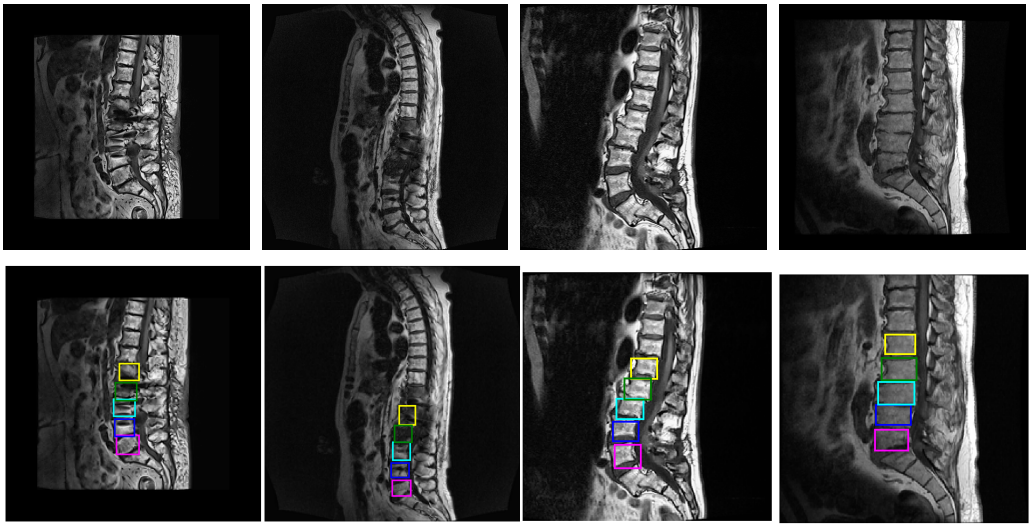
Robust localization and identification of lumbar vertebrae from spine MRI images is an essential and primary step for many clinical tasks, such as reviewing spinal

images to diagnose various spinal diseases, surgical planning, and postoperative assessment. However, it could be nontrivial and time-consuming even for human experts to localize and identify each vertebra and distinguish it from neighboring ones. The challenges include the similar appearance of neighboring highly repetitive vertebrae, different structures and shapes of each vertebra from one scan to other, narrow FOVs that makes the referencing vertebrae invisible, and no adequate resolution of scans, especially on the edges. Moreover, because of pathological cases, the anatomical shape of the vertebral column can be unpredictable when patients have spinal deformations or surgical implants around the vertebrae, which often reduces the contrast of the vertebrae boundaries. In addition, interpretation of the scans might not always be reproducible across interpreters. Therefore, devising a computational methods and establishing a computer-assisted system for automation of vertebrae localization-identification can substantially benefit the daily work of radiologists and many subsequent tasks in spinal image analysis. The focus of our study in this chapter is to address vertebrae localization-identification of the lumbar vertebrae through adopting DL techniques. We overcome challenges of this task via a straightforward DNN approach with a prior assumption regarding the number of vertebrae that appears on the scan, which is generalized with transfer learning. We formulate vertebra localization as the coordinate regression to regress corner coordinates of the surrounding boxes around every single lumbar vertebra using a very deep pre-trained network. We evaluate the performance on a heterogeneous dataset that contains a wide variety of image resolutions, different fields of view, and pathological cases. We compare our regression-based network to other approaches using encoder-decoder UNet and YOLO-based detection architectures. It is shown that our regression-based network particularly performs robustly on pathological cases, such as the spinal column images with missing vertebrae and extreme deformation caused by severe diseases or surgical implants.

5.3 Related Works

Lower back pain is one of the common problems in the general population (Wilson et al., 2021), which can be originated in the lumbar spine region. There are clinical symptoms that require investigations through computed tomography (CT) or magnetic resonance imaging (MRI) with the help of computer-assisted diagnosis.

The localization and identification of most of the clinical images are based on manual annotation by experts. For instance, for lumbar spine MRI interpretation, one needs to look through many slices of the sagittal plane and other planes of the scan to identify the correct vertebrae, which is time-consuming.



(a) (b) (c) (d)

Figure 5.1: Demonstration of some pathological cases in our dataset (the first row) and outputs of our pipeline (the second row) overlaid on those cases. The artifacts in our dataset include, (a) bright and sharp regions caused by metal implants, (b) missing vertebrae in imaging outcomes caused by spine-related severe diseases, (c) abnormal curvature and disc disorder, (d) and abnormal vertebrae shape. The second row of overlaid images shows our method's localization and identification result on the pathological cases. The surrounding bounding boxes show the lumbar vertebrae location, and the colors show the identifications. Magenta, blue, cyan, green, and yellow demonstrate the identification of L_5, L_4, L_3, L_2, and L_1, respectively.

On the other hand, these slices sometimes do not have enough length to incorporate all the necessary information. Besides, the interpretations might not be reproducible across interpreters. These are the motivations of many studies to address the problem of automating vertebrae localization-identification. However, many challenges caused by pathologies, surgical implants, image artifacts, and different FOVs should be tackled to design a robust and generalized algorithm. Figure 5.1 shows the overview of the variety of the pathological cases in our dataset.

Studies proposed to tackle the problems associated with vertebrae localization-identification can be roughly classified into two types. The first type mainly targets

specific regions of the spine, such as lumbar vertebrae and thoracic regions (Oktay and Akgul, 2011; Ma et al., 2010), or depend on the prior knowledge about visible parts (Huang et al., 2009; Schmidt et al., 2007; Kelm et al., 2010; Zhan et al., 2015; Forsberg et al., 2017).

The second type focuses on lessening the prior assumptions or the limit of visible parts on the scans. Glocker et al. (2012) presented a regression forest method of the vertebra center points in an arbitrary field of view CT scans and a refinement step using the Hidden Markov Model. This method is likely to fail in cases with a limited FOV. To address this challenge, Glocker et al. (2013) later proposed a randomized classification forest approach. This method probabilistically classified all the voxels to the particular vertebrae, and the centroid of the predictions are estimated using the mean shift algorithm. But, they make assumptions about the shape and appearance of vertebrae, which is not realizable to pathological spinal scans. More probabilistic and model-based methods were proposed by (Kelm et al., 2013; Klinder et al., 2009).

The aforementioned approaches were trained by hand-crafted feature descriptors, which cannot encode more representative features of spinal images. Therefore, they are likely to fail in handling more complicated cases when abnormalities arise. Recently, the advances of DL techniques in the computer vision field (Long et al., 2015; Ronneberger et al., 2015; Milletari et al., 2016), encouraged the community to build automatic CT/MRI vertebrae localization and identification models based on DL methods. For instance, Chen et al. (2015) presented a joint learning model to exploit high-level feature representation with a CNN. Suzani et al. (2015) proposed a multi-variate non-linear regression to parametrize the vertebrae volume localization. In this method, the reference voxel center points were identified using the Canny edge detector, and CNN learned the displacement of the rough locations to the true ones. The estimations were finally refined by an adaptive kernel density estimation method. Recently Janssens et al. (2018) proposed DL method for 3D CT lumbar spine localization. This method localized individual lumbar vertebrae through a segmentation network, UNet (Ronneberger et al., 2015), on the estimated lumbar region. Liao et al. (2018) proposed a recurrent neural network approach to address localization-identification of vertebrae. Using a fully convolutional network, they roughly estimated the vertebrae location and then refined the estimations using bidirectional recurrent neural networks. This method may lose the global dependency among the vertebrae, which is important for vertebrae identification. To handle

both local and global information, Qin et al. (2020) proposed an end-to-end model that combine a classification and detection model with an integral regression. The current methods have already achieved acceptable performance in 3D analysis, but, the complex network models with 3D data could come at high computational costs and expenses. More recently, (Wang et al., 2021) converts and simplifies 3-D detection maps into 1-D detection signals and jointly localize vertebrae following an anatomical constraint. It is shown that, however, severe pathologies and extreme imaging conditions may still negatively impact the model's performance.

Our method proposes an end-to-end DL method on a single slice of MRI scans to robustly localize and identify lumbar vertebra, which can handle pathological cases. Inspired by Toshev and Szegedy (2014), we formulate the lumbar vertebrae localization as the coordinate regression of the bounding boxes around all five lumbar vertebrae. The regression is based on the CNN approach through transfer learning, which enables us to take advantage of very deep networks (He et al., 2016a), despite the small available dataset for training. Given an MRI scan of a lumbar vertebra, our end-to-end method robustly performs a localization and identification over five lumbar vertebrae, L_5, ..., and L_1. The only assumption in our pipeline is that the scans should incorporate the lumbar region. But it does not mean all the lumbar vertebrae should be visible on the scan. For instance, in Figure 5.1.b, L_1 and L_2 were not appeared in the scan, but the pipeline still can predict their locations and labels. We achieve a 93.72% identification rate on a heterogeneous dataset containing many pathological spine MRI scans. We also develop pipelines using state-of-the-art DL architectures that are applicable to the spine localization-identification through segmentation network (Ronneberger et al., 2015) and detection network YOLO (Redmon et al., 2016). Particularly for pathological cases, We show that our regression model results in more robust performances. We further extend our pipeline to the whole spine that can relax the assumption we made on the lumbar vertebrae region. To this end we propose to first classify the FOVs and then address localization-identification of each part separately.

5.4 Evaluation Metrics

To evaluate the vertebrae localization-identification, we propose using a precision rate that represents the model's ability to accurately label lumbar verteb-

rae by satisfying a localization criteria. Precision (P) measures the ratio of the true detected objects (true positive) to the total number of detected objects as: $P = true\ positive/(true\ positive + false\ positive)$, where $true\ positive$ (TP) and $false\ positive$ (FP) are controlled by a localization threshold T on the intersection over union (IOU). IOU measures how well the ground truth object overlaps with the model predictions. The selected vertebrae satisfying $IOU \geq T$ are selected as TP; otherwise, they are assigned to FP.

IOU is also used in this chapter to measure the performance of segmentation networks: IOU $= y \cdot \hat{y}/((y + \hat{y}) - y \cdot \hat{y})$, where y and $\hat{y}$ denote the ground truth and predicted mask, respectively.

5.5 Image Segmentation for Localization and Identification

This section first gives an introduction to image segmentation and later employs the recent advances in image segmentation to formulize the lumbar vertebrae localization-identification.

5.5.1 Image Segmentation

Image segmentation has been widely used over the recent decade in a wide variety of applications, especially due to the rise of DL techniques. In medical imaging settings such as computer-aided diagnosis and smart medicine, segmentation is considered one of the key steps towards improving diagnostic efficiency and accuracy. We further demonstrate the effect of data augmentation, human-in-the-loop and loss function to improve the segmentation network performance evaluated by IOU.

Some tasks in medical image segmentation include liver segmentation (Kavur et al., 2020) and liver tumor segmentation (Budak et al., 2020), knee bone segmentation (Ambellan et al., 2019), chronic skin lesion segmentation (Nejati et al., 2016), brain tumor segmentation (Tiwari et al., 2020), interval disc and spine vertebrae segmentation (Kim et al., 2018b; Lessmann et al., 2019), and lung segmentation (Shi et al., 2020).

Image segmentation aims to divide an image into regions with strong correlations. It was conventionally addressed by drawing boundaries around the region of interest

using edge detection, template matching techniques, statistical shape models, active contours, and machine learning (Noble and Boukerroui, 2006). With the advances of the CNN, however, the focuses has been shifted towards deep neural network techniques, which surpass state-of-the-art methods by a large margin (Siddique et al., 2021).

Fully convolutional network

A fully convolutional network (FCN) is a variation of CNN (introduced in Section 3.1), which has led to great progress in image segmentation tasks (Long et al., 2015). FCN replaces the pooling layer and fully connected layers at the top of the CNN with up-sampling operators, which is a learnable parameter. Because FCN excludes the fully connected layers, the network is no longer limited to a fixed-size input and is able to generate segmentation maps with the same size as the input samples. This aligns with the goal of semantic segmentation, which formulates image segmentation as a pixel-wise classification task, where every pixel in an image is assigned with a corresponding class label. One of the early attempts to use FCL for semantic image segmentation, shown in Figure 5.2, has been done by Long et al. (2015).

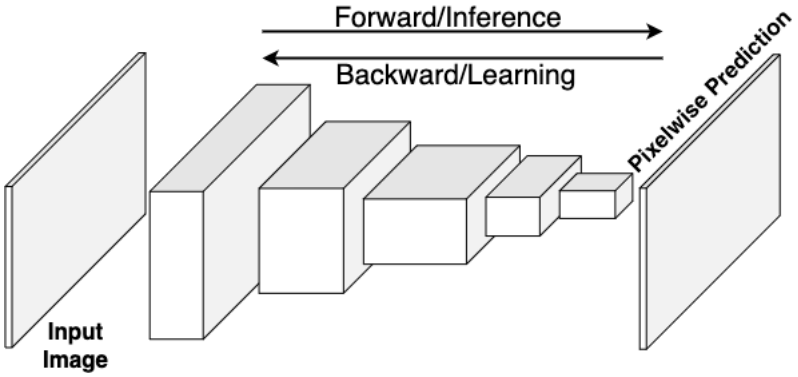


Figure 5.2: Dense predictions for semantic segmentation using fully convolutional networks. This network replaces the pooling layer and fully connected layers at the top of a CNN with learnable up-sampling operators to reach the input resolution. Therefore, through pixel-wise classification, this network outputs the input segmentation map with the same resolution as the input image.

UNet

UNet is proposed by Ronneberger et al. (2015) upon FCN architecture to solve problems of medical image segmentation such that it works precisely with very few training images to address the insufficient amount of training images. UNet modifies the up-sampling part of the FCN network by including a large number of feature channels, which allows the network to propagate contextual information to higher resolution layers. UNet has an encoder (contracting) path that transfers an image to the latent space through convolutional and max-pooling layers and a symmetric decoder (expanding) path that generates segmentation through learn-able up-sampling layers. The expansive path is designed in a way that the output of each up-sampling level has the same feature map size as the contracting pass, illustrated in (5.3). In such architecture, gradients likely vanish through propagating through many layers of the network. Subsequently, no gradient remains to update the weights, specially in the early layers of the network.

To address this problem, skip connections are added between the network layers so that high-resolution features from the contracting path can combine with the up-sampled feature maps on the expansive path. These skip connections provide a shorter path for the gradient to flow, facilitating gradient updates and convergence. Besides, long skip connections in the encoder-decoded architecture of UNet enable the recovery of the fine-grained details of images.

The U-Net has become the benchmark for most medical image segmentation tasks and has inspired numerous meaningful improvements (Lei et al., 2020). In this thesis, the application of UNet as semantic image segmentation is used to localize the spine vertebrae of MRI images.

5.5.2 Lumbar Vertebrae Localization-Identification through Segmentation

Localization

FCN architectures composed of convolutional layers can also be seen as a tool for medical data localization. Convolutional filters as learning parameters in FCN considerably degrade the number of learnable parameters, as opposed to the architectures with the same depth containing fully connected layers. This section introduces two ways of adopting FCN for lumbar vertebrae localization-identification.

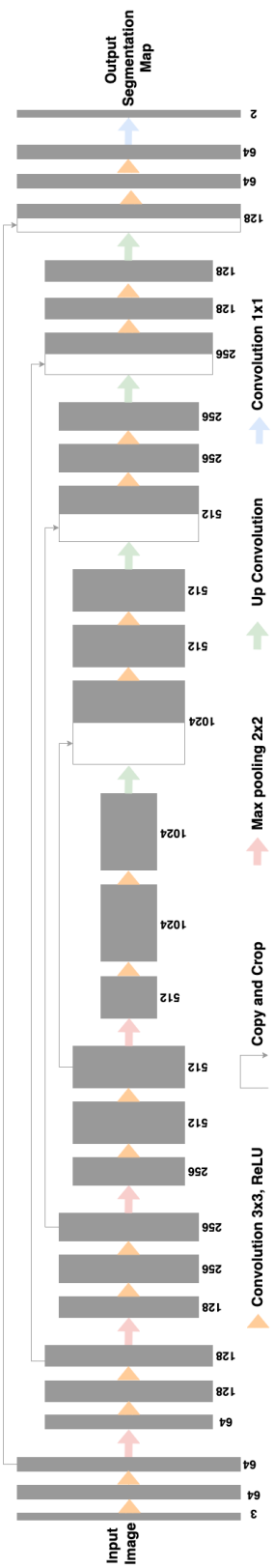


Figure 5.3: UNet architecture: convolutional neural network for biomedical image segmentation. In comparison to the fully convolutional network (Figure 5.2), UNet replaces the up-sampling part with a large number of feature channels in an expansive path. The skip connections are added to reduce the vanishing gradient problem, ease the gradient flow, and update the weight in the early layers of the network.

Network Design

We adopted UNet (Ronneberger et al., 2015) introduced in Section 5.5.1 as the segmentation network for our purpose. AS it is explained it this section U-Net contains a contracting and expanding path. The expanded layers are concatenated with the convolutional layers of the equal resolution in the contracting path. In total, our network has 23 convolutional layers, each of which are followed by a ReLU activation layer. Activation layers are followed by a 2×2 max pooling layer with stride 2 for downsampling on the contracting pass and a 2×2 transposed convolution ("up-convolution") layer on the expanding path. The role of transposed convolution is to scale the feature map to larger sizes in expanding path through convolution operations of kernels so that at the output layer, the same resolution as the input layer is obtained. Unlike up-sampling that resizes the input by copying the pixel intensities, the parameters of the transposed convolution are learned during the training phase. The number of feature channels gets halved and doubled at each down-sampling and up-convolution steps, respectively. We use small kernel size of 3×3 for all the channels. It is shown by Simonyan and Zisserman (2014) through the evaluation of networks that the smaller kernel size significantly improves the performance of the very deep networks. At the top of the network, a 1×1 convolutional layer is added to map the feature space to the desired number of classes in the output.

Training

The network is trained on MRI images as input and their masks as output. The loss function is computed by combining a pixel-wise softmax function over the final feature map with the cross-entropy loss function. Softmax activation and cross-entropy loss are defined as follows:

$$p_c(\mathbf{x}) = \exp\left(a_c(\mathbf{x})\right) / \left(\sum_{c'=1}^{C} \exp\left(a_{c'}(\mathbf{x})\right)\right), \tag{5.1}$$

$$\text{loss} = \sum_{\mathbf{x}} \log\left(p_{\ell(\mathbf{x})}(\mathbf{x})\right), \tag{5.2}$$

where $a_c(\mathbf{x})$ denotes the activation of the last layer in feature channel c at pixel position x. C denotes the number of classes and ℓ determines the true label of each pixel. $p_c(\mathbf{x})$ represent the approximated maximum function. I.e., $p_c(\mathbf{x}) \approx 1$ for the c

that has the maximum activation $a_c(\mathbf{x})$ and $p_c(\mathbf{x}) = 0$ for all other c. The loss then penalizes at each position the deviation of $p_{\ell(x)}(x)$ from 1.

We first adopt UNet, which is formulated as five class segmentation for the segmentation of five lumbar vertebrae. This enables UNet to localize and identify all lumbar vertebrae in an end-to-end manner. We refer to this approach as multi-class segmentation(M-seg). The input sample to the M-seg is the center slice of the MRI lumbar image, and the output comprises five channels. Each channel is associated with the segmentation of one vertebra body and is equal in size to the input resolution. The surrounding bounding boxes around the predicted segmentation area serve as the position of the vertebrae, and the channels determine their identification. Albeit this approach is straightforward and en-to-end, we will show that its performance is poor.

We then separate the localization and identification tasks into two stages to ease the complexity. We refer to this approach as single class segmentation (S-seg). As opposed to M-seg, we first make binary segmentation for localization of all vertebrae and then label the localized vertebrae in a post-processing scheme. Using UNet in this manner aims to perform the localization, in a way that all the vertebrae pixels (as the instances of one class) are predicted as the foreground in a single channel. Therefore, unlike optimizing the loss on five channels in M-seg, it will be run on one single channel. Our observation shows that, as it was expected, this relaxation in the training objective can improve the localization performance of the model. The following section describes how the localized vertebrae using S-seg are labeled.

Identification

Here, we describe the pipeline for labeling or identifying the vertebra in S-seg. The vertebrae in S-Seg are localized using binary masks. These masks are labeled through the following steps.

1. Clean the binary masks in the following order: (a) Fill the holes (also known as region filling) on segmented areas – where small regions inside the vertebral body are predicted as background – by machine vision morphological image processing operations. (b) Remove the small connected components whose sizes are less than 0.2 of the largest connected component. (c) Erode touching connected components horizontally for once using morphological erosion operation with 3×3 struct (a matrix of entries of one in the middle row and zeroes

otherwise).

2. Label the first connected component from the bottom as L_5. We assume that the first vertebrae localized by S-seg at the bottom of the image is L_5. Although this assumption can be violated in the cases where S-seg fails to predict the L_5 body, or it wrongly detects the sacrum bones as lumbar vertebral body, our observation shows that S-seg is highly robust in this regard.

3. Calculate the distances between the centers of the two neighboring segmented vertebrae and take the median and standard deviation of the distances.

4. Label the rest of the connected components using a criterion based on component order on the image given the reference point L_5, and the statistics calculated on the vertebrae center distances on step 3. The criterion says that if the distance of the current vertebra center from the preceding one minus the median is less than two standard deviation, assign the immediate label to the current vertebra. Otherwise, assign the second immediate label to the component. Not that we set this specific criterion for the cases that the vertebral body does not appear on the scan because of the pathological cases, and as a result, the S-seg cannot localize them.

Table 5.1: Precision rate (%) performance comparison of lumbar vertebrae localization-identification using S-seg and M-seg. The results demonstrate a significant improvement on using the segmentation network in binary mode for lumbar vertebrae detection.

Lumbar	S-seg	M-seg
L_1	75.182	45.34
L_2	71.53	45.14
L_3	59.124	49.3
L_4	64.963	60.43
L_5	80.29	75.34

Table 5.1 compares the performances of the S-seg and M-Seg. It shows the performance has been greatly improved in S-seg due to optimizing a simpler loss. This experiment show that how dividing a complex problem into simpler steps, in

practice, can boost the performance of the localization-identification precision rate. Figure 5.4 visualize some failed examples resulting from M-seg and compares them to the outcome of S-seg localization-identification.

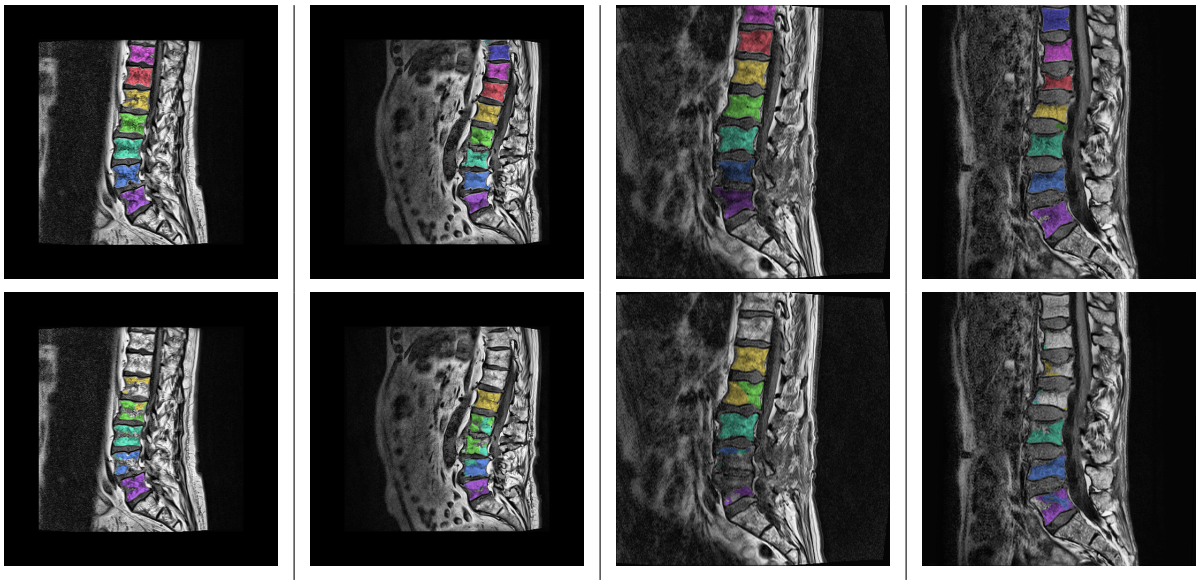


Figure 5.4: Comparison of lumbar vertebrae localization-identification through segmentation. The first row demonstrates the outcome of S-seg approach, described in Section 5.5.2, while the second row illustrates the outcome of M-seg approach, described in Section 5.5.2, on the same samples. Colors indicate the identification of vertebrae, which starts from purple as L_5 at the bottom of the spine. The colored area determines the localization performance. In quantitative evaluation, we draw a bounding box around the segmented area to measure the precision rate. As it is apparent, S-seg is more robust than M-seg in both localization and identification.

5.5.3 Data Augmentation

In this section, we aim to show the effect of data augmentation and the choice of the augmentation method on training deep convolutional networks with the focus of S-seg approach. As mentioned earlier, lack of sufficient training data leads to overfitting problems and reduce generalization ability. Besides, adversarial attack tolerance is another growing concern for researchers (Deldjoo et al., 2021). One way to address these challenges is data augmentation approaches, which synthetically generate data instances for training a model. Growing data via data augmentation makes the network more generalized and less vulnerable to adversarial attacks. This is because the network learns invariances and robustness properties without the need

to see these transformations in the annotated image corpus.

Affine Transformation It is a family of simple and easy to understand and implementation methods of data augmentation, including affine image transformation (rotation, reflection, scaling, and shearing), and color modification (histogram equalization, enhancing contrast or brightness, white balancing, sharpening, and blurring).

GANs Using Generative adversarial networks (GANs) is one of the advanced data augmentation techniques which generate new images in an unsupervised manner using min-max strategy (Engstrom et al., 2018). GANs use two adversarial networks, a generator and a discriminator. The generator generates realistic images to fool the discriminator by minimizing a cost function, while the discriminator maximizes the cost to better distinguish fake images from the real ones (Goodfellow et al., 2014). GANs are found to be useful in many image generation and manipulation problems (Creswell et al., 2018; Yi et al., 2019). Some problem, however, may arise with generating images with GANs, such as lack of compliance with reality and coordinating a global structure (Mikołajczyk and Grochowski, 2018), which is the reasons we avoid GANs for data augmentation in this section.

Elastic Deformation In addition to small affine transformations, elastic deformation (introduced by Simard et al. (2003) in the context of visual document analysis) is also known as common deformations of tissues appeared on medical images (Castro et al., 2018; Ronneberger et al., 2015; Nalepa et al., 2019). Despite studies that show performance improvements of CNNs using elastic deformation as data augmentation, there are works which indicate that such aggressive augmentation may deteriorate the performance of the models (Lorenzo et al., 2019). Therefore, the effect of excessive data augmentation using affine transformation and elastic deformation on segmentation of spinal vertebrae is investigated in this section.

In this study, scans are primarily enhanced by contrast limited adaptive histogram equalization (Yadav et al., 2014). We then excessively applied affine transformation and elastic deformation to the training set, generating 30 deformed versions for each of coupled image-masks. The deformations are selected based on the nature of the spine MRI images. For elastic deformation, we follow the setting proposed by Ronneberger et al. (2015). The smooth deformations using random displacement vectors are generated. The displacement vectors are sampled from a Gaussian distribution

Table 5.2: The effect of different data augmentation on lumbar spine segmentation using S-seg on the original dataset of 60 images. The results are reported based on intersection over union (IOU). Ten of the images are left out for the testing time, and the rest are used for training and validation sets. Augmentations are only applied to the training set. The affine transformation is primarily applied to the dataset, which improves the IOU by 17%. Then, the affine transformation further improves the results by 2.7%.

	Training(%)	Validation(%)	Testing (%)
Original data	86.42	77.28	75.30
Original data + affine transformation	96.34	94.2	92.2
Original data + affine transformation + elastic deformation	98.98	97.76	94.98

with 10 pixels standard deviation. Bi-cubic interpolation is then used to compute per-pixel displacements. The affine transformation is limited to translation of 10 pixels in x and y directions, rotation of 10 degrees, and scaling of range=[0.9,0.11].

The effect of the described data augmentations are reported in Table 5.2. As it is apparent from this table, the performance is improved by growing the data. The affine transformation is primarily applied to the dataset, which improves the IOU by 17% and degrades the overfitting (the difference between the testing and training performance) by 7%. Then, the elastic transformation further improves the results by 2.7%. From these results, it can be realized that realistic deformations reduce overfitting effect and enhance the performance of spinal vertebrae segmentation.

5.5.4 Semi-automatic Generation of Training Annotations

To alleviate the scarcity of data and provide more training annotations, we take one more step, by incorporating human expert opinion. We bootstrap the annotation with the human-in-the-loop approach (Holzinger et al., 2017; Cui et al., 2016) in which the uncertain part of the output generated with the network is modified with humans and the modified outcome is added to the training set. Not that the uncertain part of the output refers to the masks that are not completely aligned with the true regions.

In this section, we investigate whether growing annotated data in this way help the network to see more variances and consequently perform better. To this end, we

Table 5.3: Human-in-the-loop effect on the segmentation performance, IOU of S-seg. This experiment includes all the data augmentation variations introduced in Section 5.5.3. First row shows the S-seg performance on original 60 images, and the second shows the performance after one human-in-the-loop iteration. The results show an improvement of 4% in segmentation performance, and a reduction of 3% in overfitting effect (the difference in IOU performance between training and testing phases).

#Original scans	Training (%)	Validation (%)	Testing (%)
60	98.98	97.76	94.98
137	99.9	99.9	98.68

employ the S-seg approach defined in section 5.5.2 to generate preliminary results. This network is first trained on 60 original images in addition to all variants of data augmentation introduced in Section 5.5.3. The trained network is inferred on unknown samples to generate the masks over lumbar vertebrae. We know that the samples that the model perfectly annotates can not add more information to the network and are deemed redundant to be a part of training data. Therefore, we select the uncertain generated masks. Because there were no available annotations for the generated mask on the unknown samples, the uncertain samples were picked by a human expert instead of using quantitative measurements. Among the uncertain generated samples, we randomly collect part of them to be modified.

The chosen image-mask pairs are added to the training set after a single round of the human-in-the-loop cycle and two post-processing modifications. We first fill in the holes appeared on some areas over the segmented vertebrae and remove a small component over unwanted areas. Both modifications are applied using morphological operations. Then, the segmented areas are manually modified, especially the regions close to the borders. We run one iteration of this process as illustrated in Figure 5.5. This cycle can be iterated to get more masks or enhance the model over time.

As it is shown in Table 5.3 one iteration of the human-in-the-loop cycle improves the segmentation performance in all training, validation, and testing phases. Besides, the gap between training and testing performance degrades by 2%.

5.5.5 Loss Function

The choice of loss function for training efficient DL segmentation models has been experimented in various domains (Kayalibay et al., 2017; Milletari et al., 2016; Ron-

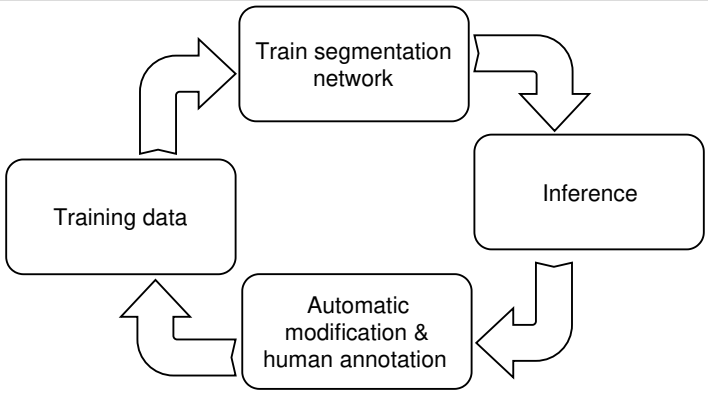


Figure 5.5: Human-in-the-loop illustration. In our pipeline, with the segmentation network as the main learning block, the network is trained on the training data, and it outputs the segmentation maps of the testing instances. From the predicted segmentation maps, the uncertain maps are selected by a human expert. Some of these uncertain maps are randomly selected and modified by automatic modification (computer vision operations) and human annotation. The modified mask and the corresponding instances are then added to the training set for the second round of training. Table 5.3 shows an improvement of 4% in segmentation performance, and a reduction of 3% in overfitting (the difference in IOU performance between training and testing phases), with one round of human-in-the-loop iteration.

neberger et al., 2015). The loss functions in DL semantic segmentation can be categorized into four groups: region-based loss, e.g., Dice and Jaccard; distributed-based loss, e.g., binary cross-entropy and balanced cross-entropy; boundary-based loss, e.g., Hausdorf distance; and compounded loss, e.g., combo. In this section, for training UNet on spine MRI images, we compare the two most commonly used losses (Kayalibay et al., 2017; Milletari et al., 2016; Ronneberger et al., 2015) in medical image segmentation: cross-entropy and Jaccard losses. Cross-entropy (Yi-de et al., 2004) measures the difference between two probability distributions for a given variable or set of events, which is widely used for pixel-level classification as a segmentation task. Binary cross-entropy loss L_{CE} is defined as:

$$L_{CE}(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})), \tag{5.3}$$

where y and $\hat{y}$ are true and predicted labels of image pixels. While this loss is shown to be successful in many applications, in a situation where the target area is rather small in comparison to the rest of the image, this loss might not reflect the actual

accuracy of semantic segmentation. When the number of pixels on the background outnumber the number of pixels on the target foreground, the accuracies of near 98% are more likely influenced by the background regions. This is the motivation to use the Jaccard loss L_{JD} that directly optimize the dissimilarity between two objects:

$$L_{JD} = -\frac{2|y| \times |\hat{y}|}{|y| + |\hat{y}|} \tag{5.4}$$

Equation (5.4) which is close to Jaccard index ($Jacc = \frac{|y \cap \hat{y}|}{|y| + \hat{y}| - |y\hat{y}|}$) is also known as IOU loss. Minimizing Jaccard loss for training segmentation network UNet minimizes the dissimilarity between the predicted mask $\hat{y}$ and the ground truth y. Therefore, the network is trained to optimize the weights so that to make $\hat{y}$ close to y.

As we mentioned earlier in Section 5.5.4, we start training UNet with the limited number of annotated samples. These samples include 60 MRI scans, on which we train the initial UNet. We train the network with similar hyperparameters with cross-entropy loss and Jaccard loss separately. Interestingly, while the training and testing segmentation performances using these two losses were comparable, the inference predictions using the network trained on Jaccard loss has shown to be far more capable on unlabeled data, which we visually observed on most of the segmented scans. This enabled us to get some preliminary segmentation masks that were later chosen as potential candidates to be added to the training set with the human-in-the-loop iteration (described in Section 5.5.4). We compare these two losses once again with 137 scans. Similar to the first try (on 60 training samples) both cross-entropy and Jaccard losses lead to equally well outcome on training and testing phases. However, this time the outperformance of Jaccard loss on unlabeled data is not as pronounced. This can be explained in two ways: (1) the remaining unlabeled data are still too difficult to learn, or (2) the added samples to the training data using human-in-the-loop, contain only redundant information to this network. In either cases, based on our experiments it can be concluded that, while Jaccard and cross-entropy loss perform equally well with more training samples, Jaccard loss can be more reliable with a smaller number of training data points.

5.6 Object Detection for Localization and Identification

The previous section presented how the localization-identification problem can be formalized by single and multi-class segmentation networks. In this section, we continue this analysis by adopting object detection networks.

5.6.1 Object Detection

Object detection aims to locate instances of semantic objects of a certain class. Object detection can be formulated by the localization of regions of interest by drawing bounding boxes around them and classification of these bounding boxes by corresponding class labels. In the medical imaging context, object detection can be applied to locate organs or disease-related regions. Let I be an image with n regions of interest. The detection function then estimates the specification of the bounding boxes, including width w, height h, the x_i and y_i coordinates of the centroid, and the class label c_i.

Traditional object detection pipelines include three stages of informative region selection, feature extraction and classification. Each stages of the traditional pipeline comes with shortcomings. Informative region were selected by scanning the whole image with a multiscale sliding window. Although this exhaustive strategy can detect all possible positions of objects, a large number of candidate windows, makes this strategy computationally expensive and inefficient in terms of memory consumption, and producing many redundant windows. Feature extractions were manually designed, such as, SIFT (Lowe, 2004), HOG (Dalal and Triggs, 2005) and Haar-like (Lienhart and Maydt, 2002) features. Due to the diversity of appearances, illumination conditions, and many other variations, these features my not be able to perfectly describe the visual features. Classification, which distinguishes a target object from all the other categories, was done using conventional machine learning models, such as, SVM (Cortes and Vapnik, 1995), Adaboost (Schapire, 2013), and Deformable Part-based Model (Felzenszwalb et al., 2010), which have limited capacity.

A significant gain in object detection has been achieved by the introduction of regions with CNN features (R-CNN) (Girshick et al., 2014). R-CNN was the first to show that a CNN can lead to dramatically higher object detection performance (on

PASCAL VOC 2010 challenge) as compared to systems based on traditional methods. R-CNN first uses a selective search (Uijlings et al., 2013) for region proposals. Proposed regions are assumed to contain the objects of interest (e.g., 2000 regions per image). The regions are then resized to a certain size and fed to CNN (due to the fixed size limitation caused by fully connected layers at the top of CNNs). The features on top of CNN enter the class-specific linear SVM for classifying the proposed regions. For example, given 20 classes in the dataset, there are 20 SVM classifiers in the pipeline. There is also one more SVM for the background classification. Besides, the bounding box coordinates offset are learned for each object class with CNN features. The main shortcoming of R-CNN is that it is computationally expensive and slow. This is because the training process consists of multiple objectives, the log loss for the softmax classifier, the hing loss for linear SVM, and the least-squares for bounding box regressions. Besides, it takes a lot of memory during training phase to optimize each stage, since these stages do not share representations. R-CNN can also be slow during testing since the features must be extracted per proposed region without sharing computation.

A number of detection improvements have been achieved by successive innovations that have been motivated by the R-CNN drawbacks. For example, Fast R-CNN (Girshick, 2015) proposed to mitigate the computational cost of convolutional operation. It runs the entire image through some convolutional layers all at once to get a high-resolution convolutional feature map corresponding to the entire image. The region proposal is then applied to this feature map, which allows us to reuse all the expensive convolutional computation across the entire image. The pooling layer then wraps the crops from feature map and runs them through fully connected layers to predict the class scores and linear regression offsets. Fast R-CNN is trained on these two losses jointly. Training Fast R-CNN is ten times faster than R-CNN because it shares all the computations between different feature maps. One bottleneck to make this method even faster is that the computational time is mainly dominated by computing the region proposals. For instance, processing all proposed regions takes less than a second, while computing region proposals using selective search takes around two seconds.

Faster R-CNN (Ren et al., 2016) addressed this bottleneck by making the network predicts its own region proposals. Similar to fast R-CNN, the entire image in this method is run through some convolutional layers to get the high-resolution feature map. Then, instead of a fixed region-proposal method, the feature map in Faster R-

CNN is run through a region-proposal network that learns to project region proposals inside the network. The remaining parts of this method are similar to Fast R-CNN, except that in faster R-CNN, the network is jointly trained with four losses. These losses include the classification and regression losses for learning the bounding boxes around region proposals, and classification and regression offsets for learning final bounding boxes.

The methods mentioned above are in the family of region-based object detection methods. There is also another family of methods for object detection, in which, rather than processing each individual potential region independently, the object detection is formalized as a regression task on the entire image, as depicted in Figure 5.6. This makes the regression-based methods quite faster than region-based methods. One of these methods is YOLO (Redmon et al., 2016) that runs a single giant CNN on images and tries to regress the bounding box coordinates and predict corresponding class labels on image grid cells. To this end, YOLO first divides the images into grid cells $S \times S$. Within those grid cells, it sets different bounding boxes, called base bounding boxes B on which the network makes different predictions, including the offsets of the base bounding boxes and the corresponding classification scores for C categories. Then, Non-max suppression selects the best fits. YOLOV2 (Redmon and Farhadi, 2017) extend YOLO by adding batch normalization to CNN layers for better generalization and allowing the grid cells to contain more than one object. In YOLOV2, to encounter the problem of complexity and accuracy, a new classification network called Darknet-19 is used as a backbone, which has 19 convolutional layers and five max-pooling layers. They remove the fully connected layers in YOLO architecture and instead add the anchor boxes to predict the bounding boxes. The updated version of this framework has then been presented in so-called YOLOV3 (Redmon and Farhadi, 2018) and YOLOV4 (Bochkovskiy et al., 2020), which has been shown to be the state-of-the-art method for real-time object detection tasks.

5.6.2 Vertebrae Localization-Identification Formulation as Object Detection

Having object detection problem introduced in the previous subsection, we now formulate lumbar vertebrae localization-identification in this concept in two ways. First, we treat each vertebra as an individual class, in which the goal of the detector is to regress and identify five different objects from different classes, L_1 to L_5. This

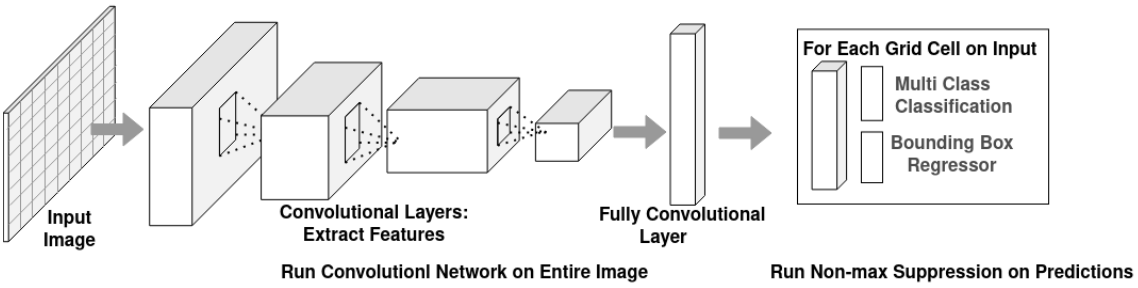


Figure 5.6: YOLO Object detection framework. YOLO is a regression based approach that use a single giant network to learn features and directly make predictions on grid cells on the entire input image. Non-max suppression at the final stage choose the most accurate anchor boxes.

approach is end-to-end and enables both localization and identification of vertebrae simultaneously. We refer to this approach as a multi-class detector (M-detect). Second, we consider all the vertebrae L_1 to L_5 as a single class, where the detector is aimed to regress all the vertebra body objects in a given input, regardless of their class labels. This way, the localization is performed by the detector, but the identification requires a labeling step later on the detected vertebrae, which we simply address by counting the lumbar vertebrae from a reference point L_5. We refer to this approach as a single class detector (S-detect). Each approach has its pros and cons. For instance, M-detect has the advantage of being end-to-end, while S-detect composes of a localization step by detection network followed by a post-processing step as the identification part. This may make S-detect less robust in the identification part. On the other hand, minimizing five class losses in M-detect makes its optimization more complex than the single class loss in S-detect, which may degrade the localization accuracy. This section investigates how S-setect and M-detect perform on our heterogeneous dataset.

Network Design

Now, let's construct our YOLO-based (Redmon and Farhadi, 2017) detection model in detail as the core design of both M-detect and S-detect. As previously introduced in Section 5.6.1, YOLO sees the entire image during training and test time, so it implicitly encodes contextual information about classes as well as their appearance. The training images for the vertebrae detection are divided into 13×13 grids, and

each grid predicts five anchor boxes. Each bounding box consists of 5 predictions: x, y, w, h, and confidence. The (x, y) coordinates represent the center of the box relative to the bounds of the grid cell. The (w, h) are width and height, which are predicted relative to the whole image. Finally, the confidence prediction represents the IOU between the predicted box and any ground truth box. Each grid cell also predicts C conditional class probabilities, $Pr(\text{Class}_i|Object)$. These probabilities are conditioned on the grid cell containing an object. During training, the following multiple parts will be optimized:

$$\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+\sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+\lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+\sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{ classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2 ,
\end{aligned} \tag{5.5}$$

where $\mathbb{1}_i^{\text{obj}}$ denotes the existence of an object in cell i and $\mathbb{1}_{ij}^{\text{obj}}$ denotes that the jth bounding box predictor in cell i is "responsible" for that prediction. λ_{noobj} and λ_{coord} are set to 0.5 and 5, respectively, which control the effect of gradient from cells that do contain objects. The network runs on 23 convolutional layers, where each layer is followed by batch normalization and leaky rectified linear activation, $\phi(x)$.

$$\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases} \tag{5.6}$$

In addition, the network is composed of four max-pooling layers. The convolutional part acts as a feature extraction unit on the whole image and outputs the bounding box (center coordinate) with the highest IOU (Intersection over the union of the ground truth and prediction) among five anchor boxes.

Training

We initialize the network with PASCAL VOC2007 (Everingham et al., 2007) weights and run the tuned model on our spine dataset and the inference on the test set. We train the network for about 100 epochs on the training and validation datasets. Throughout the training, we use a batch size of 16, the learning rate of $0.5e^{-4}$, and Adam optimizer with the first momentum of 0.9 and the second momentum of

0.999. To avoid overfitting, extensive data augmentation from affine transformation to elastic deformation, introduced in 5.5.3, are applied to the training dataset.

Testing

For testing, first, class-specific confidence scores are calculated for each box by multiplying the conditional class probabilities and the individual box confidence predictions:

$$\Pr(\text{ Class }_i \mid \text{ Object }) * \Pr(\text{ Object }) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{ Class }_i) * \text{ IOU }_{\text{pred}}^{\text{truth}} \quad (5.7)$$

These scores encode both the probability of the class appearing in the box, $Pr(\text{ Class }_i)$, and how well the predicted box fits the object, $\text{IOU }_{\text{pred}}^{\text{truth}}$. These predictions are encoded as a $S \times S \times (B \times 5 + C)$ tensor. For M-detect and S-detect, C is equal to five (for five lumbar vertebrae L_1 to L_5 detection) and one (for vertebrae detection), respectively. We use $S = 13$ and $B = 5$ for both M-detect and S-detect. Accordingly, the predictions are encoded as $13 \times 13 \times (5 \times 5 + 5)$ and $13 \times 13 \times (5 \times 5 + 1)$ tensors.

Non-max Suppression: As the output of the model, for each grid cell on an image, we receive five predicted anchor boxes. For final prediction and to select the most accurate anchor boxes, first, the low probability predictions are removed, and then for each of the classes, non-max suppression is run independently. Non-max suppression is a computer vision method that selects a single entity out of many overlapping entities to make sure the model detects each object only once. This method first discards the bounding boxes with low probabilities. Then, it iteratively takes a box with the largest probability and discards any remaining box that has IOU≥ 0.5 with the box predicted in the previous step.

Post-processing

While the non-max suppression stage is the final step for the M-detect approach, S-detect requires the labeling stage after the localization of the vertebrae. To this end, once the model predicts the most probable boxes as the individual vertebrae in the lumbar region, we assume that the first vertebra detected at the bottom of the given image is the last lumbar vertebrae L_5. Then, the remaining detected vertebrae

are labeled based on their order on the image from L_4 to L_1. In cases where one of the vertebrae is missed to be localized by the detector, the labeling is controlled by the mean and variance of the center distances of consecutive vertebrae that has been estimated over the training images. These distances are measured relative to the size of the images.

We compare these two approaches along with other methods of vertebrae detection-identification in table 5.4. We observed that S-detect performs a better localization performance, as we expected since the loss is simpler for optimization. However, both approaches perform equally well on detection, since M-detect compensates its localization outcome by making a better identification predictions. This is because in M-detect the identification part is also part of the learning. The comparison results will be discussed in more detail in Section 5.8.

5.7 New Regression Formulation for Localization and Identification

Despite acceptable performances of the YOLO-based detection network for lumbar vertebra localization-identification, this method fails in pathological cases where the abnormalities appear. Some of these abnormalities are depicted on Figure 5.1. For instance, on Figure 5.1 (b) or Figure 5.1 (a), YOLO based detection network failed detecting L_1-L_2 because of the visual artifacts. These failures occur because of: (1) different appearance of pathological cases on unseen data from the training data, and (2) considering vertebrae as individual objects with no dependency between them. To address these problems, we ease the training by introducing prior knowledge that is specific to lumbar vertebra scans. Unlike object detection task as the objective of the YOLO, which we do not have any prior knowledge about the existence of the objects in the scans, in vertebrae localization-identification task, we primarily know that scans are taken from the patient lumbar region, and therefore, five lumbar vertebrae exist on the scans, and locate in a specific order. Therefore, we make this assumption that given an MRI scan from the vertebrae region, firstly, all five vertebrae L_5 to L_1 are existed, but they may not be appeared on the scan. Secondly, they locate in descending order from the bottom of the image. Considering these assumptions, we train a deep CNN whose neurons at the output layer are fixed to regress the five lumbar vertebrae locations. The neurons' order is associated with

vertebrae identification. In this way, the network is forced to detect all five vertebrae even if they do not appear in the image like Figure 5.1 (b), or lose the order like Figure 5.1 (a). Following in this section, we formulate the vertebrae localization-identification as a regression model. We then describe how the model is optimized in the training phase and inferred in the testing phase.

5.7.1 Model formulation

We formulate lumbar vertebrae localization-identification as coordinate regression of two diagonal corners of bounding boxes around every single vertebra, using a feed-forward deep neural network for the regression problem. We train a CNN as function $\varphi(I;\theta)$ to regress vector of coordinates $y = (..., y_{i1}^T, y_{i2}^T, ...)^T$, $i \in \{1, ..., 5\}$, on a given image. θ denotes the parameters of the model, and (y_{i1}, y_{i2}) pair contains x, y absolute coordinates of upper left and bottom right corners of i^{th} lumbar vertebra. As these coordinates are assigned to specific neurons of the network, the localization and identification will be predicted at the same time. The function φ is based on a CNN structure, including a feature extraction block and regression. The feature extraction part is the very deep ResNet with 50 layers, excluding the classification decision layer. All the convolutional layers are followed by a batch normalization (BN) (Ioffe and Szegedy, 2015b) and a rectified linear activation layers (ReLU)(Nair and Hinton, 2010). The regression part is a fully connected (FCN) layer of 20 neurons, which is added to the top of the feature extraction block. Despite the other layers, FCN layer is activated linearly to enable the network to regress the coordinates in $x \in [0, img_w]$ and $y \in [0, img_h]$ intervals, in which img_w and img_h are image width and image height, respectively. We optimize the loss on the single vector y including all five lumbar vertebrae coordinates. One advantage of the defined target vector y is that we forced the network to learn the coordinates of all vertebrae, even if they do not appear on the scan. Therefore, the network is able to localize the missing vertebrae without adding more complexity to the network. To this end, our labeling strategy incorporates every lumbar vertebra, whether they are visible or missing. The missing vertebra locations in training phase are measured by the average of the precedent and subsequent vertebrae coordinates. Since the network learns the structure of missing objects on training, it will be able to predict similar cases on unknown samples. These vertebrae are omitted (Toshev and Szegedy, 2014) or considered as do-not-care (Redmon et al., 2016) in other approaches.

5.7.2 Training

We pre-train our convolutional layers on the ImageNet 1000-class competition dataset (Russakovsky et al., 2015). The choice of pre-trained weight will be discussed later in Section 5.8.4. For the pretraining, we use the first 50 convolutional layers, followed by an average-pooling layer and a fully connected layer. The whole φ is then retrained using L_1 loss, with respect to the model parameters θ as in equation 5.8, where θ^* refers to the optimal model parameters. The error is measured in absolute coordinates, and this is the reason that we rather train the network on L_1 loss than L_2 loss (used by Toshev and Szegedy (2014) and Janssens et al. (2018) for a similar regression purpose) since L_1 loss is more robust when the error is larger than 1.

$$\theta^* = \arg\min_{\theta} \sum_{i \in [1,10]} \|y_i - \varphi_i(I, \theta)\| \tag{5.8}$$

The retraining is based on mini-batch gradient descent using the Adam optimization algorithm (Kingma and Ba, 2014). To minimize the overhead and make maximum use of the GPU memory, we reduce the batch to two images. Therefore, we use a high momentum (0.99) such that a large number of the previously seen training samples determine the update in the current optimization step. The optimization starts with the learning rate of $1e^{-4}$ and weight decay of 0.1 for 100 epochs. The data augmentation, including the affine transformation and elastic deformation, is applied over the training, increasing the number of training annotations by the factor of 30.

5.7.3 Testing

The pipeline has a straightforward inference. One single image is fed to the network, and the network outputs the coordinates of the two diagonal corners of all lumbar vertebrae. The advantage of our method is that it does not need post-processing steps to refine the predictions, unlike the other recent works based on the DNN approach described in Section 5.3. To tackle the problem that different FOV may cause instead of adding more complexity to the network or the pipeline (Lessmann et al., 2018; Janssens et al., 2018), we grow the dataset in such a way that includes the FOV representing the ROI along the different FOV to the training set through excessive data augmentation and human-in-the-loop process.

One assumption of our regression-based method is that the scans contain all five

lumbar vertebrae, and therefore, the output of the network is fixed to regress the five bounding boxes. But in some cases, the scan does not cover all five lumbar regions; for instance, the scan is zoomed in for magnification. In such cases, the network gets confused about how to locate all five lumbar vertebrae on the scan unless proper padding is applied to the images. The padding can be roughly adjusted based on the number of vertebrae presented in the scan with respect to the size of the image. The rough number of vertebrae is either asked from the user or automatically estimated using the S-detect approach presented in Section 5.6.2 and then the scan is padded with zeros to include the left-out vertebrae. Besides, to make the network familiar with these cases, we simulate such data from the training set. To this end, on random samples, we crop the data to eliminate a couple of vertebrae and pad these regions with zeros.

5.8 Results

In this section, we compare the performances of the proposed methods for the localization-identification of lumbar vertebra.

5.8.1 Dataset

We evaluate the proposed method on the dataset consisting of 305 MRI scans of patients with different types of pathologies. The scans are T1-weighted, and they are a stack of sagittal slices taken from one shoulder to another. In general, the middle slice, the closest image to the middle of the vertebral column, has a higher probability of showing well-clustered vertebrae. Other slices are important for the manual annotation since they might have some information, e.g., clear edges or indications like rip bone location, making the vertebrae easier to be differentiated. Although the slices from one patient are taken from the same FOV, they vary widely through different patients, which increases the difficulties of accurate vertebrae localization-identification. This challenge has been addressed by adding augmented data, which is described in section 5.5.3. There are various pathological cases in our dataset, such as the abnormal curvature, shape, and appearance of vertebrae caused by severe diseases. Besides, the images contain different visual artifacts caused by surgical implants and the patient's movement during MRI acquisition. The whole dataset consists of two parts. There are 168 images with segmentation masks on the lumbar

vertebrae that had been grown with the factor of two by the human-in-the-loop process described in section 5.5.4. These masks are directly used as the training data for the segmentation network described in section 5.5.2. For the detection and regression networks, the surrounding bounding boxes from the segmentation mask are extracted as the ground truth. The 168 images are fed into networks for the training and validation with a split ratio of 0.2. The remaining 137 scans, annotated with the lumbar vertebrae bounding boxes, are used to compare the methods for lumbar vertebrae localization-identification. The scans in the test set are considerably more challenging in terms of pathological cases and image artifacts.

5.8.2 Experimental Results

Table 5.4 shows the precision rate of detected vertebrae with threshold $T = 0.5$ or $IOU \leq 0.5$, which is typically considered as a decent outcome in object detection problems (Redmon et al., 2016; Ren et al., 2015). In Table 5.4, we compare the outcome of all presented methods in this chapter, including, S-seg, M-seg, S-detect, M-detect, and our regression model. The results show that our regression model works significantly better than state-of the-art detection models that have been employed for the lumbar vertebrae detection. To further evaluate the results, we plot the precision rate curve generated by different threshold T in [0.1,1] interval with step size 0.05 in Figure 5.7. In this plot, S-seg and S-detect, which outperform the M-seg and M-detect respectively, are represented as the segmentation and detection based models. As it can be clearly seen, our model shows more stable outcome for all values of T on most of the vertebrae except for L_2 that S-detect surpasses for $IOU > 06$. Nevertheless, for $IOU = 0.5$, an acceptable detection rate, our method wins. These plots illustrate that our approach makes a better localization than other approaches, even when we increase the ground truth and prediction overlaps.

The average run-time per scan of size 224×224 using our method was 12.3 ms, which is excluded the time required for loading the image, overlaying the predictions on the image, and saving the results. This time is considerably less than 115 ms, which the detection network needs for vertebrae localization-identification. Using the segmentation network, the average run-time per scan of size 512×512 was 121.75 ms, comprising the segmentation part (44.108 ms) and the labeling (77.651 ms).

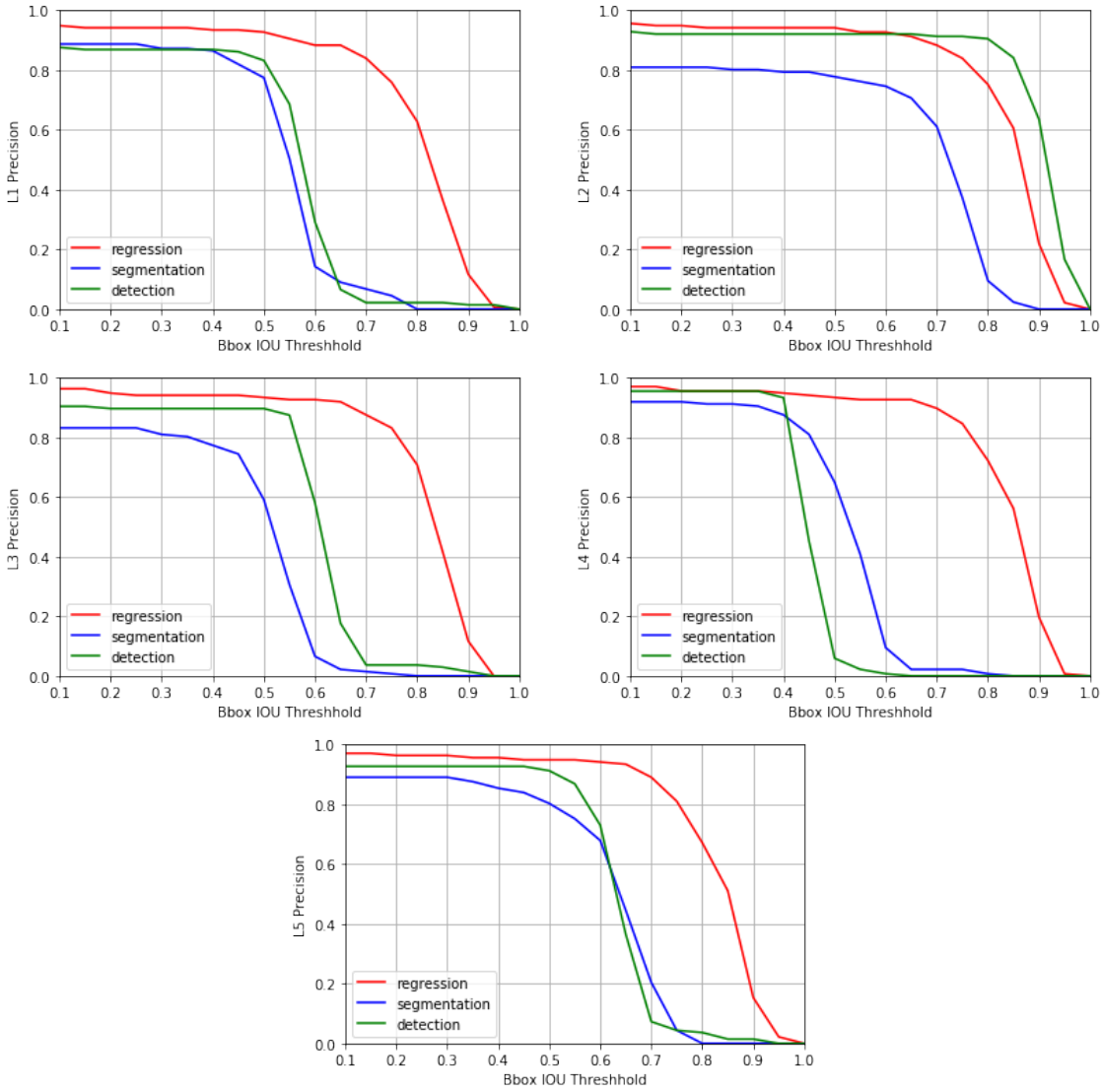


Figure 5.7: L_1 to L_5 precision rate curve generating by T between 0.1 and 1 with the step size of 0.5. The blue, green, and red lines represent s-seg, s-detect, and our regression method, respectively. The curves show that our regression method consistently outperforms the other two methods, except for L_2 that s-detect surpasses for $IOU > 06$. Nevertheless, for $IOU = 0.5$, an acceptable detection rate, our method always wins.

Table 5.4: Precision rate (%) of each individual lumbar vertebra identification on 137 test data. The values show how many of the detected vertebrae are correctly detected while the localization criteria of $IOU = 0.5$ is satisfied. Our regression method outperforms the segmentation and detection network on a test set including many pathological cases.

Lumbar	Regression	M-seg	S-seg	M-detect	S-detect
L_1	92.70	45.34	75.182	83.21	83.94
L_2	94.16	45.14	71.53	92.45	91.45
L_3	93.43	49.3	59.124	89.05	89.05
L_4	93.43	60.43	64.963	75.83	77.37
L_5	94.89	75.34	80.29	91.24	91.24
Average	93.72	55.11	70.21	84.8	85.25

5.8.3 Sensitivity Analysis

We evaluate the sensitivity of our regression model through assessing the prediction robustness to the noise. We add incremental noise to the testing data and measure the mean absolute error on regressing the two diagonal corners of bounding boxes. A zero-mean Gaussian noise generated by the different standard deviation (std) between 0.1 and 0.9 with step size 0.05 was added to the test data. The variance of a given image is defined as the average of the squared deviation of all pixels from the local mean, calculated in a two-by-two window. The signal-to-noise ratio (SNR) is defined as the ratio of the squared of image variance σ_{image} to the standard deviation of added noise σ_{noise}.

$$\sigma_{image} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(I_k - \overline{I_{local}})^2} \tag{5.9}$$

$$SNR = \frac{\sigma_{image}}{\sigma_{noise}} \tag{5.10}$$

Figure 5.8 shows the robustness of the method's prediction performance to the added noise. It is apparent from the plots that the predictions remain robust for large noise levels. As expected from a robust model, for $SNR \leq 1$, which means until the noise level becomes as large as the signal level, the network performs as accurately as when no noise is added.

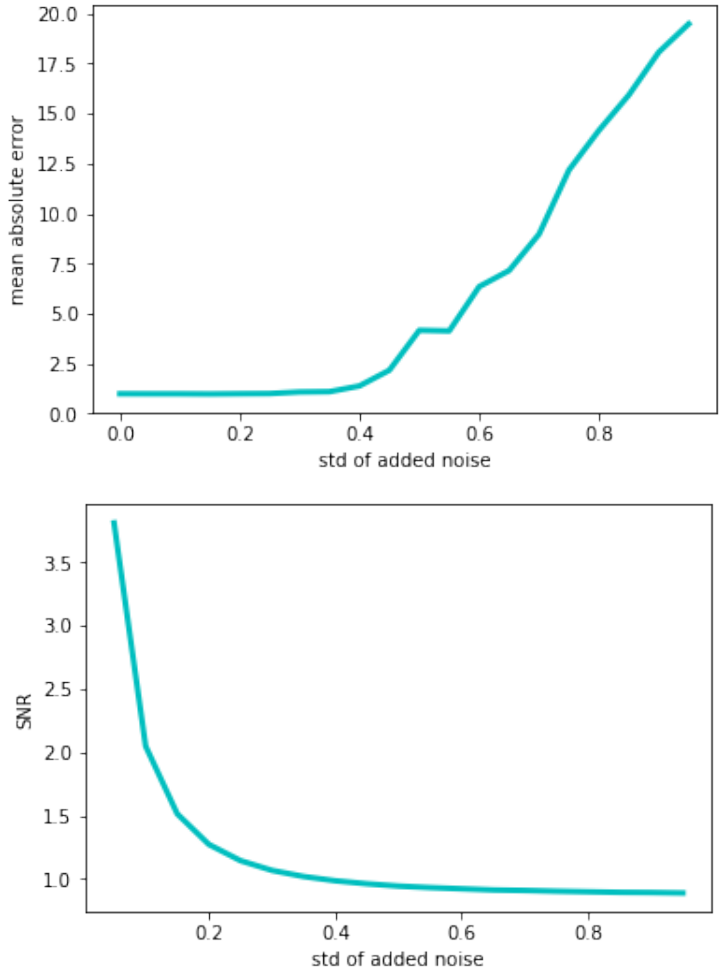


Figure 5.8: Illustration of noise robustness of the proposed regression-based network. The upper row shows the network predictions mean absolute error by increasing std of noise on a subset of test data, while the bottom row shows the changes in the SNR of those data. Approximately, until the noise level becomes as large as the signal level, the network performs as accurately as when no noise was added.

5.8.4 Network Generalization through Transfer Learning

The deep convolutional layer, in general, needs large-scale data to reach a (local) minimum that makes their parameters unbiased to the training images. However, insufficient labeled data for medical image DL analysis that has existed for years has been leading the developments to use narrower networks (Sekuboyina et al., 2017) or training the same network iteratively on subsequent stage representations (Toshev and Szegedy, 2014). On the other hand, several studies take advantage of

Table 5.5: Weight initialization impact of a data in the source domain, which has similar high-level semantic to the spine dataset, compare to weight initialization using ImageNet weights. The CT-Liver dataset is inherently similar to the spine MRI dataset, and we acquire the weights by training the whole regression network on the data, with the same goal of regressing the bounding boxes around the liver on the scans. But for the ImageNet dataset, we used the pre-trained weights available by Keras for the first 50 convolutional layers of the network. This experiment aims to show that pretraining on a dataset that barely has high-level semantic overlap with target objects also achieves equally good transfer performance for our purpose of localization identification.

#Weight initialization	Training (%)	Validation (%)	Testing (%)
Random	97.8	96.5	95.45
ImageNet	99.9	98.9	98.68
CT-Liver	99.9	98.9	98.68

well-trained, very deep networks on large-scale datasets (Yosinski et al., 2014). We used the later technique, which is called transfer learning, to generalize the proposed regression model.

Transfer learning and domain adaptation refer to the situation where what has been learned in one setting is exploited to improve generalization in another setting (Goodfellow et al., 2016). The idea is to take the representation of a neural network that has been learned from one task and transfer that representation to a new task. Please see Section 3.3 for a gentle introduction to transfer learning and its different types.

Transfer learning can also be seen as a proper weight initialization of the core part of a network that is going to be employed for a new task. In deep networks with many convolutional layers and different paths through the network, a good initialization of the weights is important. Otherwise, parts of the network might give excessive activations, while other parts never contribute (Ronneberger et al., 2015). Moreover, since the early layers of the network extract the detailed features of the input data, which are rather fixed in different kinds of image sources, proper initialization prevents the objective function from getting trapped in the local minimum or even saddle points and eventually eases the optimization. In other words, when the parameters are initialized with values close to optimum values, the error function in early epochs will get rather small, and the objective function will tend to reach minima in the correct direction. This is the reason that a pre-trained network also

has the property of becoming generalized faster. These are the reason we employ transfer learning nearly on all the networks that have been developed in this thesis for better performance and generalization.

Now, let's discuss what kind of datasets should be used for pretraining in a certain task. In a downstream paradigm, a supervised model is typically pre-trained on a large dataset of similar high-level semantics, for example, ImageNet for object recognition, which often results in a significant performance improvement. But, Zhao et al. (2021) declared that pretraining on a dataset that barely has high-level semantic overlap with target objects also achieve equally good transfer performance for object recognition problem. They showed that pretraining a network on a face dataset (VGGFace2 (Cao et al., 2018)), or a scenes' dataset (Places (Zhou et al., 2017a)) could achieve equally good transfer performance as ImageNet dataset for object recognition problems. Therefore, supervised transfer learning models mainly focus on transferring low-level and mid-level features, but not high-level features. To confirm this idea on the medical images, we compared our regression model when pre-trained on CT liver (Bilic et al., 2019) and ImageNet datasets (Russakovsky et al., 2015). The feature extraction part of our model φ is initialized with the well-trained weights of ResNet-50 on ImageNet dataset for object detection task and CT-Liver dataset for regression task. The last layers in the source domain are obviously discarded and replaced with the layer which is specialized for our regression problem, and it is initialized randomly. For transferring learning, one can either fine-tune some parts of the network by focusing on the latest layers, or retrain the whole network. Based on our experiments on a couple of networks, including segmentation, detection, and regression, retraining the whole network on target data that is semantically different from source data will lead to more robust results, especially when the number of instances in target data is small. This experiment is performed on the part of the spine data as the target that exclude pathological cases, to clearly observe the impact of transferring knowledge of different sources. Table 5.5 reports the results. The training set in the target domain contain 60 samples. The training data is separated into validation and training with a ratio of 40/60. The training samples are augmented excessively as described in Section 5.5.3. For inference, 30 new samples are used for evaluation. Results show that both datasets reach equally well-transferring knowledge, which is aligned with the study by Zhao et al. (2021).

5.9 Spinal Vertebrae Localization-Identification

In this section, we show how far we can extend the current work on lumbar vertebrae to be applicable to the whole spine, including the lumbar, thoracic, and cervical regions. Despite few training samples of those new regions, we will show how developing robust models in one region can be extended to other regions.

As we discussed in Section 5.7, to address the vertebrae localization-identification, we assumed our scans contain the lumbar region. To relax this restricted assumption, we extend the pipeline so that it first makes classification predictions to classify different FOVs, and then addresses the localization and identification of each group separately. The variation of scans' FOV in our dataset is depicted in Figure 5.9: the scans looking into the lumbar region shown on the left image, the ones that cover the larger region where all vertebrae are included on the middle image, and the scans looking into the upper region of the spine on the right image. Ideally, the spine MRI should have one more category, where scans only cover the thoracic area, but this is out of the scope of our study. For the identification of such samples, one would need indications on other plains of the MRI images to correctly distinguish different vertebrae.

For the first two groups, localization-identification of vertebrae in our pipeline is addressed by extending our regression approach developed for lumbar vertebrae. However, as mentioned, one limiting assumption of this work is that the scans are assumed to contain only five lumbar vertebrae. Therefore, the network's output is fixed to regress the five bounding boxes, which is simultaneously the key point to benefit from the rich representation of CNNs. To extend this work to an arbitrary number of vertebrae (from five to 24), one could train the regression network for all possible numbers of outputs, which is obviously not computationally efficient. Instead, we keep the regression network as a strong reference point for the lumbar region. For the remaining vertebrae, we merge the other methods to be attached to the lumbar vertebrae regression outcome. On the third group, the localization-identification can also be addressed similarly by developing a regression model for the cervical vertebrae. However, in this group, we can only address the localization since no expert identification annotations are given in our dataset to reliably study this region. Especially that first and second cervical vertebrae may not always appear or distinguishable on a slice of the sagittal plane, which makes it even more challenging

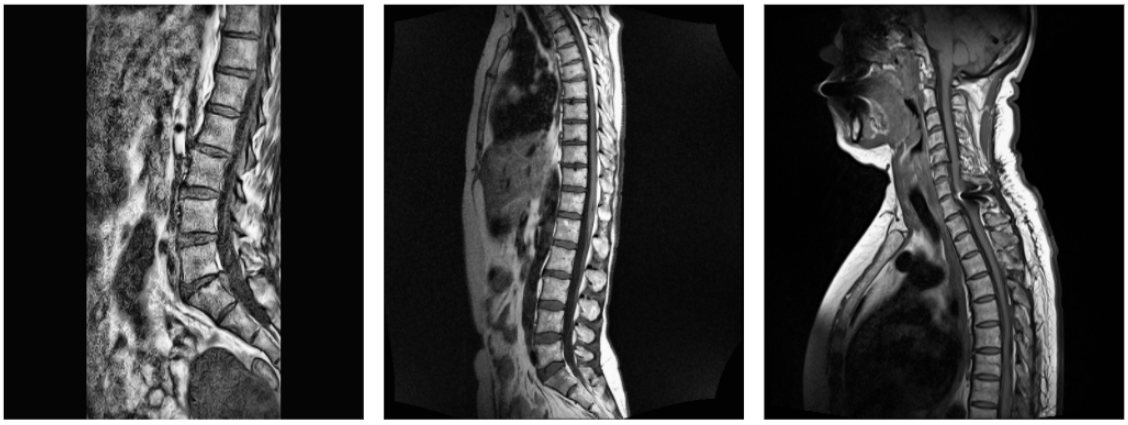


Figure 5.9: Illustration of different FOVs examples in our dataset on sagittal plane of spine MRI scans. We classify the dataset into these three FOVs and address the vertebrae localization-identification of each class separately.

for experts. For the localization of the third group, we adopt S-detect rather than the regression method due to the flexibility of S-detect in the number of output. It is worth mentioning that the main reason to use the regression network for the other two classes is to introduce a robust reference point for the identification parts. The detailed pipeline will be explained in the following sections.

5.9.1 Pipeline

As already mentioned in the previous section, the first problem with spine MRI vertebrae identification is the variant of FOV of images. To ease this complexity, we first break down localization-identification of spinal vertebrae to easier tasks by categorizing the dataset into the majority of FOVs groups, and then, the vertebrae localization-identification is addressed to each FOV separately. In the following, we describe our pipeline in detail, where again, the localization is defined as finding Vertebrae surrounding bounding boxes, and identification is defined as labeling those detected bounding boxes.

First Stage: FOV classification

Define Classes: We first classify scans into three following FOV subgroups:

- The scans show the lower part of the spine, focusing on the lumbar vertebrae.

This group that we refer to as group A comprises 97.8% of the training set.

- The scans cover lumbar, thoracic, and cervical vertebrae, which we refer to as group B.
- The rest of the scans covering the cervical region that is referred to as group C.

In more general cases, there could be another subgroup as well, in which the scans cover thinner regions, for example around thoracic vertebrae.

To get an overview of the dataset, we introduce how the scans and the annotations are distributed over different classes in the following. Note that this dataset mostly consists of group A on which essentially our task is initially defined. But, still, it is interesting to show how far one can extend a pipeline with the limited number of data points.

Training Data: There are 140 images with annotations (localization plus identification) in total in our dataset. 137 images in training-set are in group A and other 3 images in group B. There is no annotation for the cervical region. Still, we will later explain how the localization is addressed for this group as well (in Section 5.9.1). 11 images in total, consisting of 10 images from group A and one image from group B, are left out for quantitative testing to evaluate localization-identification of vertebrae.

Additional Test Data: There are 191 images without annotations in this dataset, on which we applied the whole pipeline to explore the performance of the model visually. 178 images belong to group A, 10 images to group B and 3 images to group C. The classification performance of the pipeline can be evaluated on these samples since we know FOV class annotations. But, the localization-identification of these samples is only visually observed. Figure 5.12 demonstrates the output of the pipeline on pathological cases in this dataset.

Classification Now, we describe how this dataset is classified into three classes. To classify the dataset, a tuned and regularized ResNet50 is derived and trained on the training set. But, since the majority of the scans in the training dataset comprises group A, we are facing an imbalanced classification. In this case, the model tends

to ignore samples in minority classes to achieve good performance, which results in overfitting. There are different ways to tackle the imbalanced classification problem, such as adding more weights to the minority classes, oversampling the samples in these classes, or under-sampling from the majority classes. We first modify the class weights in the loss function so that the model pays more attention to examples from the under-represented classes B and C. The weight applied to each class is $\frac{1}{num\text{-}samples} \times total/3$, where *num-samples* and *total* define the number of samples in the respected class and the total number of samples in the training dataset, respectively. Our experiments show that the weighting approach slightly improves the performance in our case. To get a better balance of data, we further added more samples of group B and C by adding scans, which we extract from other slices of DICOM images than just the middle slice. We preprocess these scans and add them to the training data along with all the variations of data augmentation described in Section 5.5.3. Note that these samples are just used for training of this stage of the pipeline, which improves the performance of the training phase significantly. On the test data, which contains 202 test samples (191 additional test data plus 11 test samples), we achieve accuracy of close to one for FOV classification.

Interpretation We interpret the network predictions to make sure our classification network works as expected. Figure 5.10 shows the interpretation of the classification network using layer-wise relevance propagation (LRP) on the respected samples given in Figure 5.9. As it is apparent from the interpretation in this figure and also based on our observation of the majority of samples in the test set, the network follows a similar pattern to distinguish the samples within one group. For instance, the interpretation of group A prediction highlights the lower part of the spine, which is the sacrum area, as the most important area for network prediction. This area for us as a human is also an apparent discriminating area to distinguish lumbar vertebra. Interpretations of group B predictions show the focus of the network on thoracic regions. Detecting this region is also aligned by the human conception for discriminating this group. For group C the interpretations highlight different regions such as the very lower region of the spine, the middle part, and the curvature of the back of the neck. These may not be the first region or reason that we as humans deem discriminating, rather, the first discriminating feature for us most probably is the size of the cervical vertebra, which is significantly smaller in this region. But, this does not mean we cannot make decisions based on other discriminating features,

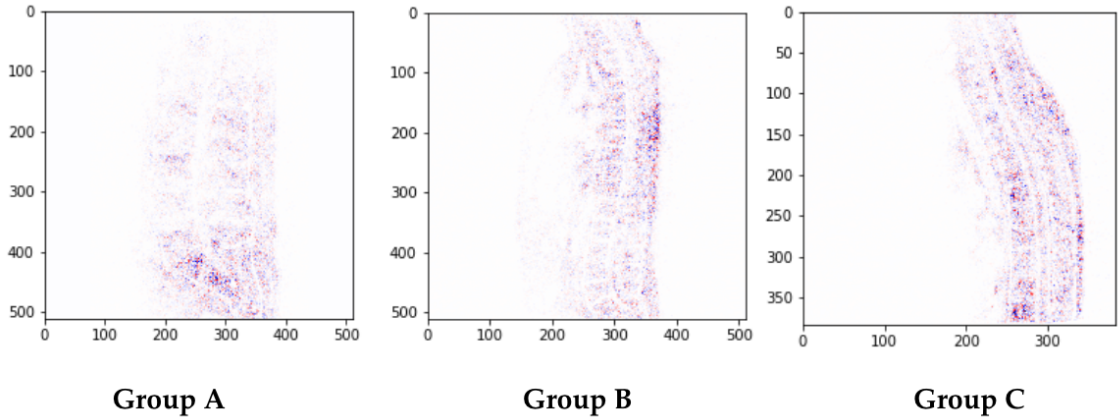


Figure 5.10: Interpretation illustration of categorizing fields of view of spine vertebrae. These visualizations show the classification interpretation respected images in Figure 5.11. We use layer wise relevance propagation for interpreting the decisions. This interpretation shows which pixels are more important for the network to classify images into three categories. As it is apparent on the right, the network sees the lower part of the spine as discriminating areas to classify the images as group A. In the middle, the interpretation shows the middle of the spine as the discriminating area. On the left, different regions such as the very lower region of the spine, the middle part, and the curvature of the back of the neck are discriminating for the network to make decisions. These explanations are reasonable since they are comparable to the way a human discriminates the three defined classes, especially for group A and B. For Group C, the size of the vertebrae can be more discriminating for human, but it does not mean decisions can not be made based on the other discriminating features.

which are found by the network and are still reasonable. Using the interpretation of the classification network, we verify that the model reliably classifies the samples based on true features.

Slice Selection: There are certainly more indications to identify different vertebrae on the other slices of the sagittal plane. However, our goal is to build a model using one slice of the sagittal plane. In our pipeline, this slice is chosen from the middle slice of entire scans of DICOM images since it has a higher probability of showing well-clustered vertebrae. In some cases, for instance, when patients are affected by severe diseases, this consideration may not hold, resulting in lower performances. In these cases, a couple of slices in the middle, two slices from the left and two slices from the right, are shown to the user. Then, the user chooses the best

slice among the shown ones.

Second Stage: Vertebrae localization and identification of group *A*

Group A covers vertebrae of the lower region of the spine, comprises lumbar vertebrae and a couple of lower thoracic vertebrae as well (Figure 5.9, left image). To address the localization-identification of this group, we employ and extend our model for robust lumbar vertebrae localization-identification, introduced in Section 5.7, by merging its results with the results from the detection network for localization of the rest of the vertebrae. We kept the model, as it is, for the lumbar region as a strong reference point, and the rest of the detected vertebrae are labeled in a post-processing. The Following list describes step by step of this process in detail:

1. Localize and identify the lumbar vertebrae using a CNN that we developed to regress the coordinates of the bounding boxes, described in section 5.7. The bounding boxes around the vertebra are regressed by the output of the specific neurons, which at the same time identify their labels.

2. The rest of the vertebrae above the lumbar region are localized using S-detect, a YOLO-based object detection approach, introduced in Section 5.6.2. We adapted the network to detect all the visible vertebrae in the image.

3. The vertebrae detected by S-detect, which are located above the lumbar region, are kept as thoracic vertebrae and are labeled in post-processing using a criterion based on their order and $L_1 - L_5$ as the reference points. In cases where the localization misses a vertebra, a term based on a statistic is added to this criterion for modification.

4. For each detected component as thoracic vertebrae, we calculate the distances between the centers of the last five consecutive vertebrae centers and take the median and standard deviation of the distances.

5. In the post-processing, the criterion says that if the distance of the precedent vertebra center from the previous one minus the median is less than two standard deviations, assign the immediate label to the precedent vertebra. Otherwise, assign the second immediate label to the component and fill the missing vertebrae by the mean coordinates from the precedent and previous vertebrae bounding boxes.

Third stage: Vertebrae localization and identification of group *B*.

Samples in group B comprise larger FOV than group A. Therefore, the vertebrae and interval disc are considerably smaller in this region. Although the pipeline developed for group A could be employed on group B as well, the small and compacted components in group B degrades its performance. Moreover, few training samples in this class make it more difficult to learn these small components, despite applying excessive data augmentation. These hinder the S-detect to localize the vertebrae robustly. However, the localization results of S-seg, introduced in Section 5.5.2 demonstrate less sensitivity to the problems mentioned above. In this stage, we still keep the regression network for lumbar vertebrae localization-identification similar to Stage 5.9.1. The difference is that the rest of the components above the lumbar region is localized and identified using S-seg as follows:

1. Localize and identify the lumbar vertebrae using the regression network (similar to the second stage 5.9.1, step1).

2. Localize the rest of vertebrae using S-seg (introduced in Section 5.5.2.)

3. The results from the S-seg are post-processed with morphological operations to remove the unwanted small areas, fill out the holes in the segmented components, and separate the adherent vertebrae: (a) Fill the holes (also known as region filling) on segmented areas – where small region inside the vertebral body is predicted as background -, by machine vision morphological image processing operations; (b) Remove the small size connected components which sizes are less than 0.2 of the largest connected component; (c) Erode the touching connected components horizontally using morphological erosion operation, for once, with 3×3 struct – a matrix entry of 1 in the middle row and zeros elsewhere.

4. The segmented vertebrae above the lumbar region are labeled based on their order on the image, the reference points $L_1 - L_5$, and a statistic of distances between consecutive vertebrae, similar to the second stage 5.9.1, step 4 and 5.

Fourth stage: Vertebrae localization of group *C*.

The samples in group C cover the upper region of the spine, the cervical vertebrae, in which we address the localization task. Since we had no training annotations for

this group, we generated rough localization annotations for 60 scans using S-detect that has been trained on samples of group A and B. Then, we modify the annotations manually. To address the localization of this group, we again employ S-detect as the core architecture of the model. This model is pre-trained on all training samples of group A and B. To enrich the feature space, we added model-based features such as the Canny edge detector as extra information to the network since the vertebrae in this region are small, and edges might not always be clear. Besides, we add a low representation of the S-seg that has been trained on the entire training dataset to one of the layers of the S-detect in the middle of the network to mount the performance. The entire network is then re-trained with this additional information. Enriching the features space as well as pre-training the model on other FOV's of vertebrae help alleviate under-fitting in the training phase and generalization performance. Given a sample classified as group C, the pipeline to localize the vertebrae is as follows:

1. Run the sample through S-seg and extract the low representation of the segmentation network, where the encoder reaches the end of the pass. Then, add this representation as an input layer to the middle of the S-detect with the same filter size.

2. Extract the Canny edges of the given image and add this information to the input layer from the top of the network, where the original image is fed to the network.

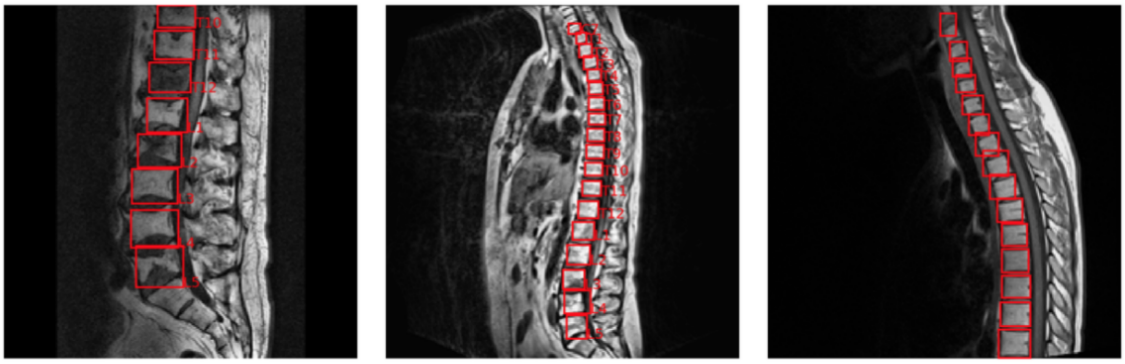
3. Localize all the visible vertebrae using the adapted network.

4. Post-proccess the results similar to second stage 5.9.1, step 4 and 5.

We observe that, on the test-set of three samples C, we localize all the visible vertebrae.

5.9.2 Results

We achieved a precision rate of 96% on average for localization-identification of all vertebrae, including 11 images in the test set from group A and group B. Besides, our collaborators and we visually confirm the robustness of the pipeline on the samples of the additional test set. Figure 5.11 shows examples of the output of the pipeline for three categories. In Figure 5.12, we depict how well the pipeline works on pathological cases.



Group A **Group B** **Group C**

Figure 5.11: Spine vertebrae detection pipeline output. The pipeline for Group A and B localization-identification and Group C localization is presented in Section 5.9.1.

5.10 Discussion

This chapter aimed to further our study of high-dimensional biomedical data using deep neural networks in an imaging modality. We investigated the challenges of this analysis and the possible solution strategies in the context of MRI data interpretation. We addressed the challenging problem of lumbar vertebrae localization-identification on human Spine MRI in which similar to proteomics data analysis, we predominantly dealt with the scarcity and heterogeneity of the data. We developed a new pipeline for lumbar vertebrae localization-identification using DL strategies and extended our approach to the whole spinal vertebrae. Towards this goal, various imaging tasks, including classification, segmentation, regression, and detection, were investigated.

We formulated lumbar vertebrae localization as the coordinate regression of surrounding bounding boxes around each individual vertebrae. Based on the presented dataset, containing 137 testing instances with a variety of pathological cases, our pipeline achieved almost 94% precision rate on average for localization-identification of five lumbar vertebrae. We showed that our regression approach significantly outperforms the other methods that have been studied in this chapter for solving lumbar vertebrae localization-identification, including variations of UNet and YOLO networks. The superior performance is more highlighted in challenging pathological spine MRI cases because we employ prior knowledge in our assumptions regarding

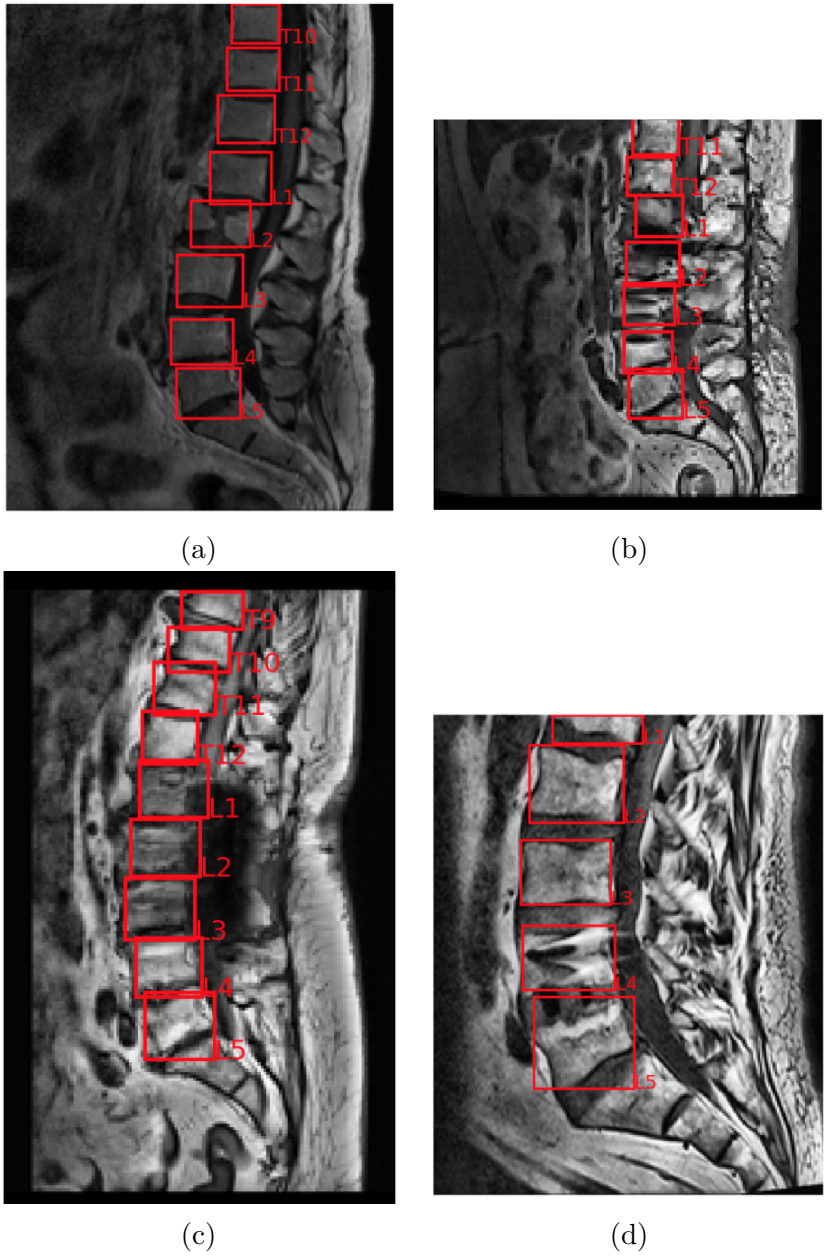


Figure 5.12: The output of our pipeline on pathological cases. The presented pipeline is able to address localization-identification of a wide variety of vertebrae deformation, including (a) two separated bodies belonging to one vertebra, (b) no clear edges and poor appearance, (c) difference color intensities between vertebrae in one scan, and (d) brighter regions caused by implants.

the fixed number of vertebrae on the scans. This assumption can also help predict the vertebra body even when it does not appear in the image. The main strength of the proposed regression method for lumbar vertebrae detection is that it has an end-to-end pipeline that does not need extra steps to refine the predictions.

Further, we proposed a pipeline that extends the study of lumbar-vertebrae to the whole spinal vertebrae that relieved the restricting assumption on the lumbar region. We first eased this task by dividing the detection problem into subgroups, in which the scans are first classified into major FOVs, and each subgroup is analyzed separately. On the presented test set containing 11 instances, our pipeline attained a precision rate of 96% on average for localization-identification of all vertebrae. Notice that the classification of FOVs was explained in our study using LRP interpretation, where we showed and explained that the interpretations of classification predictions were aligned by true discriminating areas.

In addition, through a comprehensive study of different CNN topologies in detection problems, we elucidated their strength and weaknesses. We adopted UNet, a segmentation network, and YOLO, a regression-based detection network, for our localization-identification of lumbar vertebrae. We first adopted UNet in a single class model to segment all lumbar vertebrae and use the surrounding bounding boxes as their localization. We then identified a reference point and ordered the localized vertebrae to address the identification part. In addition, we adopted UNet in multi-class segmentation as well, which enables localization and identification of the vertebrae simultaneously. We showed that the earlier approach that we called S-seg led to significantly better results. We continued the localization-identification analysis of lumbar vertebrae using DL detection networks with the aim of enhancing the outcome that had been obtained by the segmentation-based approaches. We adopted the YOLO network as a state-of-the-art approach for this purpose, again in two modes. First, it was employed to regress coordinates of bounding boxes around each vertebra as instances of one class for the localization part. Then, the identification, similar to segmentation, is addressed by identifying a reference point and labeling the rest of the detected bounding boxes by counting. We referred to this approach as S-detect. Second, like the segmentation approach, we employed YOLO to regress the surrounding vertebrae bounding boxes and label them simultaneously, which we referred to as M-detect. We empirically showed that although S-detect performs better at localization since the loss is simpler for optimization, M-detect compensates for its localization weakness by doing a better identification performance since the identific-

ation part is also part of the learning process. We demonstrated both M-detect and S-detect perform equally well and better than S-seg at localization-identification of the lumbar vertebra. Nonetheless, our end-to-end regression-based network outperforms the pipelines based on segmentation and detection network on lumbar vertebra localization-identification.

Towards tackling the scarcity of labeled data, We demonstrated the effect of the human-in-the-loop process, data augmentation, and transfer learning in improving system accuracy. By adopting proper data augmentation, in particular affine transformation and elastic deformation, the inference IOU for segmentation of the region of interest has improved by 20%. Further, it is shown that with one human-in-the-loop cycle, the performance can be increased by 4%. Using transfer learning, we also showed an increased performance by 3%. Thereby, we mainly investigated the weight initialization of contrastive models for the model improvements. We experimentally elucidated that the source data for pretraining a network do not necessarily need to be semantically similar to the target data. As for training on the lumbar vertebra, initializing the network on the ImageNet (containing classes for object detection)reached equally good transfer performance as initializing the network on the CT-Liver image dataset that has more similar semantic to medical imaging in our problem.

The accurate generalization performance using our method has important implications for developing a DNN localization method on computer-assisted diagnoses. Especially since we study different aspects of imaging analysis, including classification, segmentation, regression, localization, identification, and detection. These studies convey an in-depth understanding of high-throughput medical imaging analysis.

One of the limitations and also future work to this study is the investigation of the scans, which consist of narrow FOVs, where an indication or a reference point for starting the labeling cannot be easily found even for human experts by just looking into the one slice.

Chapter 6

Discussion, Conclusion, and Outlook

High-throughput datasets are pervasive in medical research. Analyzing these data effectively and efficiently is deemed essential in providing reliable support for developing interpretable clinical decision support systems. This thesis has introduced new methods for analyzing high-throughput biomedical data, employing modern machine learning techniques through tackling high-dimensionality and scarcity of data. This final chapter gives concluding remarks and suggest possible future directions.

6.1 Discussion and Conclusion

Medical data analysis is aimed to find patterns and extract information with the purpose of facilitating disease diagnosis, prognosis, and treatments. When it comes to high-throughput data analysis, several challenges immediately arrive, such as scarcity, high-dimensionality, and complexity of data. This thesis dealt with these challenges in a variety of high-throughput medical data analyses by adopting modern machine learning concepts, including classification, feature selection, model explanation, image segmentation, and object detection.

In the first part of our analysis in chapter 3, we addressed the challenges associated with the classification of high-throughput proteomics data whose samples contain more than 50000 features that are required to be studied to discover biological relevant information. Such large-scale data leads to the sparsity problem, limiting conventional machine learning models to fit robustly. In addition, the scarcity of data worsen this limitation due to the overfitting problem. To ease this problem, machine learning methods are equipped with dimension reduction steps, which can raise the risk of losing biological relevant information. This phenomena is likely to happen in proteomics data due to the high order of magnitude of different peak intensities and the high noise content of data, which may drop critical relevant information. Another limitation of conventional machine learning methods is their limited capacity, which is certainly not suitable for large-scale data analysis.

To overcome these limitations, we proposed a deep learning-based classification method on high-dimensional structured data. Deep learning benefits from scalable capacity, and transferring the knowledge between tasks or domains. The scalable capacity of deep learning enables enhancing the performance by growing the data, which might not be feasible in medical settings due to expensive data acquisition and annotation. In such case, deep learning can be equipped with transferring the

knowledge also known as transfer learning. The idea is to take the knowledge that the neural network has learned from one task as a source and apply that knowledge to a separate task as a target, which enables deep learning to benefit from its capacity. We employed this idea for proteomics data classification of healthy and disease classes. Since finding good-quality source data could also be infeasible in many real-world scenarios, we proposed to synthetically generate the source data that must share similar characteristics with the target data. The source and the target data may not share the high-level features, but they should share similar low-level representations in a similar context. For instance, in our study, to learn the representation of MALDI-MS human proteomics, the synthetic data were generated from the human protein complex in the MALDI-MS simulator. We generated the synthetic data so that the samples in two classes are representative of healthy and disease groups. It is realized that when training the proteomics data on a network that initialized with synthetic data weights, the outcomes became more robust and reproducible across different runs and different data batches. We confirmed this observation in MALDI-MS and LC-MS proteomics real data classification.

Built on this robustly trained network, we developed a feature selection model to extract biomarkers – disease relevant biological information – from high-throughput proteomics data. We aimed to learn from the trained network and find what makes the network arrive at a certain decision. In the diseased/healthy classification task described earlier, if we discover which features are deemed more important for the network to make certain decisions, we could potentially discover biomarker candidates. It has been a common belief that simpler models provide higher interpretability compared their complex counterparts. However, this belief is challenged by recent works, in which carefully designed interpretation techniques have shed light on some of the most complex and deepest machine learning models (Zhang et al., 2021; Huang et al., 2020). Thus, deep learning is no longer considered a black box, which enables us to extract the important features, for instant, through studying the model's prediction. We built our biomarker detection model on a backpropagation interpretation strategy called Layer wise relevance propagation (LRP). We successfully showed that a robustly trained deep neural network classifier coupled with a proper interpretation strategy could reveal the underlying disease-relevant pattern of the data. The real data analysis with our approach results in the smallest set of biomarker candidates, which can surpass state-of-the-art conventional machine learning performances.

We also compared different interpretation strategies, such as input×gradient,

input-gradient, SmoothGrad, Deconvolution, and guided-backpropagation methods. As a result, we demonstrated that the performance of LRP on highlighting the true discriminating regions are slightly better than methods like input×gradient and input-gradient, but it is significantly better than other methods, such as, SmoothGrad, Deconvolution, and guided backpropagation. These results can be explained by the fact that LRP takes advantage of structured layers of neural networks and simplifies the explanation problem. Besides, despite gradient-based methods that are locally calculated, LRP considers the whole picture of the input, in addition to being less prone to discontinuity effect.

We demonstrated that in comparison with well-established prior works (msInspect, MZmine 2, Progenesis, and XCMS), our method enables the discovery of biomarkers with more sensitivity and a lower false-negative rate. Furthermore, this success was achieved while we skipped expensive preprocessing steps often employed in conventional methods. We ran the experiments on raw data to avoid loosing relevant information through data preprocessing steps.

In the second part of our analysis in Chapter 4, the application of deep learning explanation was further investigated for high-throughput data analysis through quantitatively assessing the prediction interpretations. Suppose that we know which features of the data are relevant for making decisions. In this case, the model can be tuned, or its robustness can be measured by checking if the interpretation of the model is aligned with the relevant features. Now, assume that different network architectures with different depth and number of layers for a classification task share similar accuracy and performance. To decide which architecture to choose, we can check which of these networks results in interpretations that align with the relevant features. But, in real-world tasks, especially when dealing with high-throughput data, the annotations on relevant features in this level are unknown or could be very expensive to acquire. In such cases, we suggested selecting the architecture based on synthetically generated data, in which we know about the relevant features. We then demonstrated that this information regarding the network architecture is transferable to real data analysis, assuming that the synthetic data shares similar characteristics with the data being studied. We successfully showed that the network that is primarily tuned on synthetic data is more robust, and its interpretation is more aligned with the true, relevant features.

One important key that should be examined prior to incorporating the deep learning interpretation is to check their trustworthiness and reliability, which we

demonstrated through sensitivity analysis for repeatability, reproducibility, and consistency of the outcomes. In our interpretation-based biomarker detection study, we investigated three criteria: 1) whether the interpretations of the same samples are reproducible when the network is trained with random seeds, 2) whether the interpretations of the same samples present a repeatable pattern when the network is trained on a different subset of data, and 3) whether the interpretations are consistent across different samples within one class when the network is trained on one or a different subset of data. In these criteria, intersection over union (IOU) between the interpretations was considered as the similarity measure. For instance, in the second criterion, we reached IOU near to one between interpretations of test sample's predictions which are made by networks trained on the different folds of data (in five-fold cross-validation mode). We observed a similar performance on other criteria, as well, which provides strong evidence to support the trustworthiness and reliability of our interpretation-based approach.

The aforementioned quantitative assessments, however, can be done when the annotations at the feature level is known or can be synthesized. Otherwise, the interpretations can be assessed through heuristic approaches. In our work, biomarkers were selected using the network prediction interpretations. Therefore, we reformulated the interpretation assessment to the biomarker detection assessment. Then, we measured the importance of the selected biomarkers in the classification accuracy. We performed this experiment on three real MALDI-MS data. The results illustrated that the model reaches almost the maximum accuracy with only first few selected high-ranked features, elucidating the high quality of the network interpretations.

In the third part of our analysis in Chapter 5, other machine learning concepts, including segmentation, detection, and regression, were investigated. We changed our focus from high-dimensional structured proteomics data analysis to analyzing imaging data modality with the application of localization of regions of interest. In early chapters, for the purpose of localization of biological relevances, we used network interpretation which enables the discovery of unknown patterns from the data. In the third part, with a change of direction, we investigated the supervised deep learning approaches. We aimed for the challenging spinal lumbar vertebrae localization and identification task, where we formulated object detection with different convolutional neural network topologies, including segmentation and detection networks in binary and multi-class modes. We found that simplifying the multi-class tasks to the binary mode that is coupled with an extra refinement step can significantly contribute to

the generalization performance in heterogeneous data.

As previously stated, a major obstacle to the translation of data-driven technologies to clinical settings is the lack of good quality data, where we showed a significant generalization improvement with the human-in-the-loop process, adding proper data augmentations, and the right choice of the objective function. We then proposed a robust new regression-based pipeline to localize individual vertebrae, focusing on the lumbar region that successfully performed better than prior works. It especially works well on pathological cases, mainly because we assumed to have a fixed number of vertebrae in an image. With this assumption, we were able to anticipate the location of regions that do not appear on the scan due to artifacts and severe diseases. Based on the presented heterogeneous dataset, containing 137 testing instances, our pipeline achieved almost 94% precision rate on average for localization-identification of five lumbar vertebrae.

All the networks in our downstream tasks were primarily trained with medical images of other tasks for transferring the representation, which eases the optimization. But, we also showed that the source domain dataset does not necessarily require having high-level semantic with the target domain. This means, for transferring the representation in deep learning, the low-level features are deemed beneficial transferable information from one task to another. This observation is previously realized in natural object recognition applications, in which we provide evidence in the medical imaging domain.

Through a comprehensive study of different CNN topologies, we elucidated their strength and weaknesses, which provides a valuable understanding of high-dimensional medical imaging interpretation using modern machine learning techniques. Built on our observation, we finally proposed a pipeline that extends the lumbar-vertebrae to the whole spinal vertebrae interpretation by dividing this complex task into simpler subgroups. This pipeline results in a 96% precision rate and has been actively used by the project partner. Our approach is

To summarize, this thesis has addressed two important questions related to high-throughput data analysis, which were considered at the beginning of this thesis.

1. High-dimensionality and scarcity of data. How to deal with the curse of dimensionality and limitation of good quality labeled data?

High-dimensionality and scarcity of data are the major topics of this thesis, owing to the expensive process of standard medical data collection and annotations. In

Chapter 3, we tackled this issue on structured data by employing transfer learning. We showed pre-training a DL model on a huge synthetically generated data that share similar characteristics with the target task results in a robust classification result. We demonstrated how we effectively take advantage of synthetic data not only for pre-training purposes but also for a better classification network design in Chapter 4, which were both then translated into real-world data analyses. We provided our findings to learn the representation of high-throughput proteomics data that severely suffers from the curse of dimensionality and contains a high level of noise. In Chapter 5, addressing the scarcity of data was discussed in high resolution medical imaging data analysis. We demonstrated the effect of data augmentation and human-in-the-loop to alleviate the problem in the challenging task of human spinal vertebrae detection. We also studied the transfer learning, where the main concern was to show that the source data for pretraining do not necessarily need to be semantically similar in appearance to target data. We started the analysis with segmentation application, and then successfully extended it all over this chapter for classification, detection, and regression purposes, as well.

2. Learn data patterns from the machine. How to improve the transparency of deep learning models through interpretability as it potentially leads to a better understanding of the data and the deployed model?

As the second concern in this thesis, we considered the role of deep learning interpretability in analyzing high-throughput biomedical data. In Chapter 3, we showed that when we build a robust DNN classifier, for example, for disease prediction, we can learn from the prediction interpretations where the network looks into the data for discriminative patterns. This information enables discovering the biomarker candidates and associating them with specific diseases. To validate this claim, we provided evidences regarding the robustness of the explanations. We exemplified our findings with the application of biomarker discovery in high-throughput mass-spectrometry proteomics data, which surpassed conventional biomarker discovery pipelines. Prior works were highly dependent on consecutive steps with several parameter tuning that may differ from one setting to another. But, we showed with a minimal human intervention we can yet outperform prior works. We also demonstrated that without a single label at the biomarker level, and just by learning medical states on training data, the biomarker candidates can be estimated accurately. We extended our interpretability analysis in Chapter 4 through a series of

assessments that serve as feedback to the system. We employed this feedback to enhance the model performance by adjusting the parameters of the model developed for biomarker discovery, such as architecture design. Furthermore, we employed the interpretability assessment to better understand the model, which provides evidence for the reliability and robustness of the model. The application of interpretability has also been employed in Chapter 5 in the context of imaging modality with the aim of confirming classification decisions as a part of an image analysis pipeline.

6.2 Future Directions

Finally, we suggest potential future research directions related to high-throughput data analysis.

- Self-supervised learning: Our analyses on structured tabular data in Chapter 3, and imaging data in Chapter 5 indicate the significance of transferring the knowledge between domains to help the network convergence and generalization ability. But in many applications, the target dataset contains more unlabeled data than labeled ones, like our spine MRI dataset in Chapter 5, and the question is that whether the knowledge from unlabeled data could further improve the model performances. Self-supervised learning (Jing and Tian, 2020) is one possible and recent way of extracting the representations from unlabeled data. This strategy learns the representations similar to supervised learning through optimizing on inputs and labels. The difference is that the labels in this strategy are generated automatically by machine based on a predesigned task, without involving any human annotation. This representation is then used as initialization weights in downstream tasks by fine-tuning. An example for learning visual features is *SimCLR* (Chen et al., 2020) approach that learns representations by maximizing agreement between differently augmented views of the same training example using a contrastive loss. It is shown that *SimCLR* outperforms some models, e.g., AlexNet with 100x fewer labels. It is interesting to study whether a similar strategy can be adapted in medical data analysis that severely suffer from lack of labeled data.

- Transformer: Attention-based networks have been widely adapted to many ML domains, such as natural language processing (NLP) (Wolf et al., 2020),

computer vision (Khan et al., 2021), speech recognition (Dong et al., 2018), etc. Attention in transformer architecture was born to resolve losing dependencies in long sequences with sequence-to-sequence models in language modeling. But, the powerful properties of the transformer have encouraged scientists to explore its implications in many domains as well. This require reformulating tasks into into sequencing problems. For instance, to take advantage of the transformer in machine vision tasks, the images are converted into sequences of patches, embedded to vectors, similar to the word embedding in NLP. A recent study has shown that emerging the self-supervised properties in the transformer can potentially let the attention map accurately highlight the relevant information in the image (Caron et al., 2021). It has been shown that the learned representation can be used in a variety of applications, e.g., in down stream tasks for fine-tuning, image retrieval, zero-shot classification through K-nearest neighbor classifier, and image segmentation. With this powerful representation from unlabeled data, it is quite interesting to see the implication of transformers in the context of high-throughput medical data analysis that can be meaningfully converted to sequence analysis. It is especially important to see if the attention map could highlight relevant information on complex biological data.

- Multi-modality: In a clinical setting, physicians may diagnose diseases according to reviewing different examinations that might be in different modalities, e.g., medical imaging, certain features in blood samples in the form of tabular data, and metadata written in the text about the patient symptoms, etc. Multi-modal machine learning is one way to implement this idea. This approach aims to build models that can process and relate information from various aspects (Baltrušaitis et al., 2018). This idea can be coupled with the self-supervised learning approach to understanding diseases. (Li et al., 2020) showed that learning representation in a self-supervised format via exploiting multi-modal data results in comparable performance to the supervised format. This approach was described with the application to Retinal disease diagnosis, in which the second modality was synthesized. It is interesting to employ this idea for omics data analysis for biomarker detection in Chapter 3 for better performance and a smaller amount of data.

Bibliography

Abadi, Martín et al. (2016). 'Tensorflow: A system for large-scale machine learning'. In: *12th Symposium on Operating Systems Design and Implementation*. USENIX Association, pp. 265–283.

Abràmoff, Michael D et al. (2018). 'Pivotal trial of an autonomous AI-based diagnostic system for detection of diabetic retinopathy in primary care offices'. In: *NPJ digital medicine* 1.1, pp. 1–8.

Aebersold, Ruedi and Matthias Mann (2003). 'Mass spectrometry-based proteomics'. In: *Nature* 422.6928, p. 198.

Agarwal, Chirag, Dan Schonfeld and Anh Nguyen (2019). 'Removing input features via a generative model to explain their attributions to classifier's decisions'. In: *arXiv preprint arXiv:1910.04256*.

Akkus, Zeynettin et al. (2019). 'A survey of deep-learning applications in ultrasound: Artificial intelligence–powered ultrasound for improving clinical workflow'. In: *Journal of the American College of Radiology* 16.9, pp. 1318–1328.

Albaradei, Somayah et al. (2021). 'Machine Learning and Deep Learning Methods that use Omics Data for Metastasis Prediction'. In: *Computational and Structural Biotechnology Journal*.

Alber, Maximilian et al. (2019). 'iNNvestigate neural networks'. In: *Journal of Machine Learning Research* 20.93, pp. 1–8.

Ambellan, Felix et al. (2019). 'Automated segmentation of knee bone and cartilage combining statistical shape knowledge and convolutional neural networks: Data from the Osteoarthritis Initiative'. In: *Medical image analysis* 52, pp. 109–118.

Anwar, Syed Muhammad et al. (2018). 'Medical image analysis using convolutional neural networks: a review'. In: *Journal of medical systems* 42.11, pp. 1–13.

Aoshima, Ken et al. (2014). 'A simple peak detection and label-free quantitation algorithm for chromatography-mass spectrometry'. In: *BMC bioinformatics* 15.1, p. 376.

Arrieta, Alejandro Barredo et al. (2020). 'Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI'. In: *Information Fusion* 58, pp. 82–115.

Arun, Nishanth et al. (2020). 'Assessing the (un) trustworthiness of saliency maps for localizing abnormalities in medical imaging'. In: *arXiv preprint arXiv:2008.02766*.

Bach, Sebastian et al. (2015). 'On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation'. In: *PloS one* 10.7.

Bairoch, Amos et al. (2005). 'The universal protein resource (UniProt)'. In: *Nucleic acids research* 33.suppl_1, pp. D154–D159.

Baltrušaitis, Tadas, Chaitanya Ahuja and Louis-Philippe Morency (2018). 'Multimodal machine learning: A survey and taxonomy'. In: *IEEE transactions on pattern analysis and machine intelligence* 41.2, pp. 423–443.

Bau, David et al. (2017). 'Network dissection: Quantifying interpretability of deep visual representations'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549.

Bellew, Matthew et al. (2006). 'A suite of algorithms for the comprehensive analysis of complex protein mixtures using high-resolution LC-MS'. In: *Bioinformatics* 22.15, pp. 1902–1909.

Berner, Eta S (2007). *Clinical decision support systems*. Vol. 233. Springer.

Bilic, Patrick et al. (2019). *The Liver Tumor Segmentation Benchmark (LiTS)*.

Bochkovskiy, Alexey, Chien-Yao Wang and Hong-Yuan Mark Liao (2020). 'Yolov4: Optimal speed and accuracy of object detection'. In: *arXiv preprint arXiv:2004.10934*.

Breiman, Leo (1996). 'Bagging predictors'. In: *Machine learning* 24.2, pp. 123–140.

Buchlak, Quinlan D et al. (2020). 'Machine learning applications to clinical decision support in neurosurgery: an artificial intelligence augmented systematic review'. In: *Neurosurgical review* 43.5, pp. 1235–1253.

Budak, Ümit et al. (2020). 'Cascaded deep convolutional encoder-decoder neural networks for efficient liver tumor segmentation'. In: *Medical hypotheses* 134, p. 109431.

Cabitza, Federico, Raffaele Rasoini and Gian Franco Gensini (2017). 'Unintended consequences of machine learning in medicine'. In: *Jama* 318.6, pp. 517–518.

Calloway, Stacy, Hameed A Akilo and Kyle Bierman (2013). 'Impact of a clinical decision support system on pharmacy clinical interventions, documentation efforts, and costs'. In: *Hospital pharmacy* 48.9, pp. 744–752.

Calvo, Florence Quesada et al. (2009). 'Biomarker discovery in asthma-related inflammation and remodeling'. In: *Proteomics* 9.8, pp. 2163–2170.

Cao, Qiong et al. (2018). 'Vggface2: A dataset for recognising faces across pose and age'. In: *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*. IEEE, pp. 67–74.

Caron, Mathilde et al. (2021). 'Emerging properties in self-supervised vision transformers'. In: *arXiv preprint arXiv:2104.14294*.

Castro, Eduardo, Jaime S Cardoso and Jose Costa Pereira (2018). 'Elastic deformations for data augmentation in breast cancer mass detection'. In: *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE, pp. 230–234.

Challen, Robert et al. (2019). 'Artificial intelligence, bias and clinical safety'. In: *BMJ Quality & Safety* 28.3, pp. 231–237.

Chandrashekar, Girish and Ferat Sahin (2014). 'A survey on feature selection methods'. In: *Computers & Electrical Engineering* 40.1, pp. 16–28.

Chapelle, Olivier, Bernhard Scholkopf and Alexander Zien (2009). 'Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]'. In: *IEEE Transactions on Neural Networks* 20.3, pp. 542–542.

Chen, Chung-Ming et al. (2013). *Computer-aided detection and diagnosis in medical imaging*.

Chen, Hao et al. (2015). 'Automatic localization and identification of vertebrae in spine CT via a joint learning model with deep neural networks'. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 515–522.

Chen, Ting et al. (2020). 'A simple framework for contrastive learning of visual representations'. In: *International conference on machine learning*. PMLR, pp. 1597–1607.

Chen, Xiaojun et al. (2016). 'A semi-automatic computer-aided method for surgical template design'. In: *Scientific reports* 6.1, pp. 1–18.

Chollet, François et al. (2015). *Keras*. https://keras.io.

Chrástek, Radim et al. (2005). 'Automated segmentation of the optic nerve head for diagnosis of glaucoma'. In: *Medical image analysis* 9.4, pp. 297–314.

Conrad, Tim OF et al. (2006). 'Beating the noise: new statistical methods for detecting signals in MALDI-TOF spectra below noise level'. In: *International Symposium on Computational Life Science*. Springer, pp. 119–128.

Conrad, Tim OF et al. (2017). 'Sparse Proteomics Analysis–a compressed sensing-based approach for feature selection and classification of high-dimensional proteomics mass spectrometry data'. In: *BMC bioinformatics* 18.1, p. 160.

Consortium, UniProt (2019). 'UniProt: a worldwide hub of protein knowledge'. In: *Nucleic acids research* 47.D1, pp. D506–D515.

Cortes, Corinna and Vladimir Vapnik (1995). 'Support-vector networks'. In: *Machine learning* 20.3, pp. 273–297.

Cox, J and M Mann (2008). 'MaxQuant enables high peptide identification rates, individualized ppb-range mass accuracies and proteome-wide protein quantification'. In: *Nature biotechnology* 26.12, pp. 1367–1372.

Creswell, Antonia et al. (2018). 'Generative adversarial networks: An overview'. In: *IEEE Signal Processing Magazine* 35.1, pp. 53–65.

Cui, Yin et al. (2016). 'Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1153–1162.

Curry, Lynn and Martin H Reed (2011). 'Electronic decision support for diagnostic imaging in a primary care setting'. In: *Journal of the American Medical Informatics Association* 18.3, pp. 267–270.

Dalal, Navneet and Bill Triggs (2005). 'Histograms of oriented gradients for human detection'. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee, pp. 886–893.

Dash, Manoranjan and Huan Liu (1997). 'Feature selection for classification'. In: *Intelligent data analysis* 1.1-4, pp. 131–156.

Date, Yasuhiro and Jun Kikuchi (2018). 'Application of a deep neural network to metabolomics studies and its performance in determining important variables'. In: *Analytical chemistry* 90.3, pp. 1805–1810.

Deldjoo, Yashar, Tommaso Di Noia and Felice Antonio Merra (2021). 'A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks'. In: *ACM Computing Surveys (CSUR)* 54.2, pp. 1–38.

Dias, Duarte and João Paulo Silva Cunha (2018). 'Wearable health devices—vital sign monitoring, systems and technologies'. In: *Sensors* 18.8, p. 2414.

Dong, Linhao, Shuang Xu and Bo Xu (2018). 'Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition'. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 5884–5888.

Donoho, David L et al. (2006). 'Compressed sensing'. In: *IEEE Transactions on information theory* 52.4, pp. 1289–1306.

Doshi-Velez, Finale and Been Kim (2017). 'Towards a rigorous science of interpretable machine learning'. In: *arXiv preprint arXiv:1702.08608*.

Došilović, Filip Karlo, Mario Brčić and Nikica Hlupić (2018). 'Explainable artificial intelligence: A survey'. In: *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*. IEEE, pp. 0210–0215.

Duchi, John, Elad Hazan and Yoram Singer (2011). 'Adaptive subgradient methods for online learning and stochastic optimization'. In: *Journal of Machine Learning Research* 12.Jul, pp. 2121–2159.

Dupree, Emmalyn J et al. (2020). 'A critical review of bottom-up proteomics: The good, the bad, and the future of this field'. In: *Proteomes* 8.3, p. 14.

Eicher, Tara et al. (2020). 'Metabolomics and multi-omics integration: a survey of computational methods and resources'. In: *Metabolites* 10.5, p. 202.

Engstrom, Logan et al. (2018). 'A rotation and a translation suffice: Fooling cnns with simple transformations'. In.

Espadoto, Mateus et al. (2019). 'Toward a quantitative survey of dimension reduction techniques'. In: *IEEE transactions on visualization and computer graphics* 27.3, pp. 2153–2173.

Esteva, Andre et al. (2017). 'Dermatologist-level classification of skin cancer with deep neural networks'. In: *nature* 542.7639, pp. 115–118.

Everingham, M. et al. (2007). *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*.

Faca, Vitor, Hong Wang and Samir Hanash (2009). 'Proteomic global profiling for cancer biomarker discovery'. In: *Mass Spectrometry of Proteins and Peptides*. Springer, pp. 309–320.

Falcone, Paolo et al. (2007). 'Predictive active steering control for autonomous vehicle systems'. In: *IEEE Transactions on control systems technology* 15.3, pp. 566–580.

Felzenszwalb, Pedro F, Ross B Girshick and David McAllester (2010). 'Cascade object detection with deformable part models'. In: *2010 IEEE Computer society conference on computer vision and pattern recognition*. Ieee, pp. 2241–2248.

Fenn, John B et al. (1989). 'Electrospray ionization for mass spectrometry of large biomolecules'. In: *Science* 246.4926, pp. 64–71.

Fiedler, Georg Martin et al. (2009). 'Serum peptidome profiling revealed platelet factor 4 as a potential discriminating peptide associated with pancreatic cancer'. In: *Clinical Cancer Research* 15.11, pp. 3812–3819.

Fischl, Bruce et al. (2002). 'Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain'. In: *Neuron* 33.3, pp. 341–355.

Fong, Ruth C and Andrea Vedaldi (2017). 'Interpretable explanations of black boxes by meaningful perturbation'. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3429–3437.

Forsberg, Daniel, Erik Sjöblom and Jeffrey L Sunshine (2017). 'Detection and labeling of vertebrae in MR images using deep learning with clinical annotations as training data'. In: *Journal of digital imaging* 30.4, pp. 406–412.

Fortunati, Valerio et al. (2013). 'Tissue segmentation of head and neck CT images for treatment planning: a multiatlas approach combined with intensity modeling'. In: *Medical physics* 40.7, p. 071905.

Frewen, Barbara E et al. (2006). 'Analysis of peptide MS/MS spectra from large-scale proteomics experiments using spectrum libraries'. In: *Analytical chemistry* 78.16, pp. 5678–5684.

Friedman, Jerome, Trevor Hastie and Rob Tibshirani (2010). 'Regularization paths for generalized linear models via coordinate descent'. In: *Journal of statistical software* 33.1, p. 1.

García, Salvador, Alberto Fernández and Francisco Herrera (2009). 'Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems'. In: *Applied Soft Computing* 9.4, pp. 1304–1314.

Gaudet, Pascale et al. (2017). 'The neXtProt knowledge base on human proteins: 2017 update'. In: *Nucleic acids research* 45.D1, pp. D177–D182.

Ghafoorian, Mohsen et al. (2017). 'Location sensitive deep convolutional neural networks for segmentation of white matter hyperintensities'. In: *Scientific Reports* 7.1, pp. 1–12.

Ghorbani, Amirata and James Zou (2019). 'Data shapley: Equitable valuation of data for machine learning'. In: *arXiv preprint arXiv:1904.02868*.

Gibb, Sebastian and Korbinian Strimmer (2012). 'MALDIquant: a versatile R package for the analysis of mass spectrometry data'. In: *Bioinformatics* 28.17, pp. 2270–2271.

— (2015). 'Differential protein expression and peak selection in mass spectrometry data by binary discriminant analysis'. In: *Bioinformatics* 31.19, pp. 3156–3162.

Girshick, Ross (Dec. 2015). 'Fast R-CNN'. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Girshick, Ross et al. (2014). 'Rich feature hierarchies for accurate object detection and semantic segmentation'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Glocker, Ben et al. (2012). 'Automatic localization and identification of vertebrae in arbitrary field-of-view CT scans'. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 590–598.

Glocker, Ben et al. (2013). 'Vertebrae localization in pathological spine CT via dense classification from sparse annotations'. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 262–270.

Glorot, Xavier, Antoine Bordes and Yoshua Bengio (2011). 'Deep sparse rectifier neural networks'. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 315–323.

Goodfellow, Ian et al. (2016). *Deep learning*. Vol. 1. MIT press Cambridge.

Goodfellow, Ian J et al. (2014). 'Generative adversarial networks'. In: *arXiv preprint arXiv:1406.2661*.

Graves, Alex (2011). 'Practical variational inference for neural networks'. In: *Advances in neural information processing systems*. Citeseer, pp. 2348–2356.

Graziani, Mara, Vincent Andrearczyk and Henning Müller (2018). 'Regression concept vectors for bidirectional explanations in histopathology'. In: *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*. Springer, pp. 124–132.

Guan, Xiangming (2015). 'Cancer metastases: challenges and opportunities'. In: *Acta pharmaceutica sinica B* 5.5, pp. 402–418.

Gupta, Shubham et al. (2019). 'DIAlignR provides precise retention time alignment across distant runs in DIA and targeted proteomics'. In: *Molecular and Cellular Proteomics* 18.4, pp. 806–817.

Hamet, Pavel and Johanne Tremblay (2017). 'Artificial intelligence in medicine'. In: *Metabolism* 69, S36–S40.

Hancer, Emrah, Bing Xue and Mengjie Zhang (2020). 'A survey on feature selection approaches for clustering'. In: *Artificial Intelligence Review* 53.6, pp. 4519–4545.

Haq, Amin Ul et al. (2019). 'Feature selection based on L1-norm support vector machine and effective recognition system for Parkinson's disease using voice recordings'. In: *IEEE access* 7, pp. 37718–37734.

Hara, Satoshi and Kohei Hayashi (2016). 'Making tree ensembles interpretable'. In: *arXiv preprint arXiv:1606.05390*.

Hase, Peter and Mohit Bansal (2020). 'Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior?' In: *arXiv preprint arXiv:2005.01831*.

Haufe, Stefan et al. (2014). 'On the interpretation of weight vectors of linear models in multivariate neuroimaging'. In: *Neuroimage* 87, pp. 96–110.

He, Kaiming et al. (2016a). 'Deep residual learning for image recognition'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

— (2016b). 'Deep residual learning for image recognition'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Helmons, Pieter J et al. (2015). 'Drug-drug interaction checking assisted by clinical decision support: a return on investment analysis'. In: *Journal of the American Medical Informatics Association* 22.4, pp. 764–772.

Henry Vandyke Carter, Public domain, via Wikimedia Commons (1918). URL: https://commons.wikimedia.org/wiki/File:Gray_111_-_Vertebral_column-coloured.png.

Hinton, G. E., S Osindero and Y Teh (2006). 'A fast learning algorithm for deep belief nets.' In: *Neural Computation* 18, 1527–1554.

Hinton, Geoffrey et al. (2012a). 'Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups'. In: *IEEE Signal processing magazine* 29.6, pp. 82–97.

Hinton, Geoffrey E et al. (2012b). 'Improving neural networks by preventing co-adaptation of feature detectors'. In: *arXiv preprint arXiv:1207.0580*.

Hoffmann, Franziska et al. (2019). 'Identification of Proteomic Markers in Head and Neck Cancer Using MALDI–MS Imaging, LC–MS/MS, and Immunohistochemistry'. In: *PROTEOMICS–Clinical Applications* 13.1, p. 1700173.

Holzinger, Andreas et al. (2017). 'A glass-box interactive machine learning approach for solving NP-hard problems with the human-in-the-loop'. In: *arXiv preprint arXiv:1708.01104*.

Holzinger, Andreas et al. (2019). 'Causability and explainability of artificial intelligence in medicine'. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9.4, e1312.

Hou, Tsung-Yun, Chuan Chiang-Ni and Shih-Hua Teng (2019). 'Current status of MALDI-TOF mass spectrometry in clinical microbiology'. In: *Journal of food and drug analysis* 27.2, pp. 404–414.

Huang, Gao et al. (2016a). 'Deep networks with stochastic depth'. In: *European Conference on Computer Vision*. Springer, pp. 646–661.

Huang, Gao et al. (2016b). 'Densely connected convolutional networks'. In: *arXiv preprint arXiv:1608.06993*.

Huang, Szu-Hao et al. (2009). 'Learning-based vertebra detection and iterative normalized-cut segmentation for spinal MRI'. In: *IEEE transactions on medical imaging* 28.10, pp. 1595–1605.

Huang, Xiaowei et al. (2020). 'A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability'. In: *Computer Science Review* 37, p. 100270.

Huizing, Lennart RS et al. (2019). 'Development and evaluation of matrix application techniques for high throughput mass spectrometry imaging of tissues in the clinic'. In: *Clinical Mass Spectrometry* 12, pp. 7–15.

Iandola, Forrest N et al. (2016). 'SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size'. In: *arXiv preprint arXiv:1602.07360*.

Ioffe, Sergey and Christian Szegedy (2015a). 'Batch normalization: Accelerating deep network training by reducing internal covariate shift'. In: *International conference on machine learning*. PMLR, pp. 448–456.

— (2015b). 'Batch normalization: Accelerating deep network training by reducing internal covariate shift'. In: *International Conference on Machine Learning*, pp. 448–456.

Iravani, Sahar and Tim OF Conrad (2019). 'Deep Learning for Proteomics Data for Feature Selection and Classification'. In: *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, pp. 301–316.

Janssens, Rens, Guodong Zeng and Guoyan Zheng (2018). 'Fully automatic segmentation of lumbar vertebrae from CT images using cascaded 3D fully convolutional

networks'. In: *Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on*. IEEE, pp. 893–897.

Jayrannejad, Fahrnaz and Tim OF Conrad (2017). 'Better Interpretable Models for Proteomics Data Analysis Using Rule-Based Mining'. In: *Towards Integrative Machine Learning and Knowledge Extraction*. Springer, pp. 67–88.

Jia, Pengli et al. (2016). 'The effects of clinical decision support systems on medication safety: an overview'. In: *PloS one* 11.12, e0167683.

Jing, Longlong and Yingli Tian (2020). 'Self-supervised visual feature learning with deep neural networks: A survey'. In: *IEEE transactions on pattern analysis and machine intelligence*.

Jumper, John et al. (2021). 'Highly accurate protein structure prediction with AlphaFold'. In: *Nature* 596.7873, pp. 583–589.

Kantz, Edward D et al. (2019). 'Deep neural networks for classification of LC-MS spectral peaks'. In: *Analytical chemistry* 91.19, pp. 12407–12413.

Karas, Michael and Franz Hillenkamp (1988). 'Laser desorption ionization of proteins with molecular masses exceeding 10,000 daltons'. In: *Analytical chemistry* 60.20, pp. 2299–2301.

Kavur, A Emre et al. (2020). 'Comparison of semi-automatic and deep learning-based automatic methods for liver segmentation in living liver transplant donors'. In: *Diagnostic and Interventional Radiology* 26.1, p. 11.

Kayalibay, Baris, Grady Jensen and Patrick van der Smagt (2017). 'CNN-based segmentation of medical imaging data'. In: *arXiv preprint arXiv:1701.03056*.

Kelm, B Michael et al. (2010). 'Detection of 3D spinal geometry using iterated marginal space learning'. In: *International MICCAI Workshop on Medical Computer Vision*. Springer, pp. 96–105.

Kelm, B Michael et al. (2013). 'Spine detection in CT and MR using iterated marginal space learning'. In: *Medical image analysis* 17.8, pp. 1283–1292.

Kermany, Daniel S et al. (2018). 'Identifying medical diagnoses and treatable diseases by image-based deep learning'. In: *Cell* 172.5, pp. 1122–1131.

Khakharia, Aman et al. (2021). 'Outbreak prediction of COVID-19 for dense and populated countries using machine learning'. In: *Annals of Data Science* 8.1, pp. 1–19.

Khan, Salman et al. (2021). 'Transformers in vision: A survey'. In: *ACM Computing Surveys (CSUR)*.

Kim, Been et al. (2018a). 'Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)'. In: *International conference on machine learning*. PMLR, pp. 2668–2677.

Kim, Sewon et al. (2018b). 'Fine-grain segmentation of the intervertebral discs from MR spine images using deep convolutional neural networks: BSU-Net'. In: *Applied sciences* 8.9, p. 1656.

KIMME, CAROLYN, BERNARD J O'LOUGHLIN and Jack Sklansky (1977). 'Automatic detection of suspicious abnormalities in breast radiographs'. In: *Data structures, computer graphics, and pattern recognition*. Elsevier, pp. 427–447.

Kindermans, Pieter-Jan et al. (2017). 'Learning how to explain neural networks: Patternnet and patternattribution'. In: *arXiv preprint arXiv:1705.05598*.

Kingma, Diederik P and Jimmy Ba (2014). 'Adam: A method for stochastic optimization'. In: *arXiv preprint arXiv:1412.6980*.

Klinder, Tobias et al. (2009). 'Automated model-based vertebra detection, identification, and segmentation in CT images'. In: *Medical image analysis* 13.3, pp. 471–482.

Kohlbacher, Oliver et al. (2007). 'TOPP—the OpenMS proteomics pipeline'. In: *Bioinformatics* 23.2, e191–e197.

Kohlbrenner, M. et al. (2020). 'Towards Best Practice in Explaining Neural Network Decisions with LRP'. In: *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7.

Kratzsch, Juergen et al. (2005). 'New reference intervals for thyrotropin and thyroid hormones based on National Academy of Clinical Biochemistry criteria and regular ultrasonography of the thyroid'. In: *Clinical chemistry* 51.8, pp. 1480–1486.

Krizhevsky, Alex, Ilya Sutskever and Geoffrey E Hinton (2012). 'Imagenet classification with deep convolutional neural networks'. In: *Advances in neural information processing systems*, pp. 1097–1105.

Kuster, Bernhard et al. (2005). 'Scoring proteomes with proteotypic peptide probes'. In: *Nature reviews Molecular cell biology* 6.7, pp. 577–583.

Lamping, Florian et al. (2018). 'Development and validation of a diagnostic model for early differentiation of sepsis and non-infectious SIRS in critically ill children-a data-driven approach using machine-learning algorithms'. In: *BMC pediatrics* 18.1, pp. 1–11.

Larson, David B et al. (2018). 'Performance of a deep-learning neural network model in assessing skeletal maturity on pediatric hand radiographs'. In: *Radiology* 287.1, pp. 313–322.

LeCun, Y. et al. (1988b). 'Gradient based learning applied to document recognition.' In.

LeCun, Yann, Yoshua Bengio and Geoffrey Hinton (2015). 'Deep learning'. In: *nature* 521.7553, pp. 436–444.

Lei, Tao et al. (2020). 'Medical Image Segmentation Using Deep Learning: A Survey'. In: *arXiv preprint arXiv:2009.13120*.

Lessmann, Nikolas et al. (2018). 'Iterative fully convolutional neural networks for automatic vertebra segmentation'. In: *arXiv preprint arXiv:1804.04383*.

Lessmann, Nikolas et al. (2019). 'Iterative fully convolutional neural networks for automatic vertebra segmentation and identification'. In: *Medical image analysis* 53, pp. 142–155.

Li, Xiaomeng et al. (2020). 'Self-supervised feature learning via exploiting multi-modal data for retinal disease diagnosis'. In: *IEEE Transactions on Medical Imaging* 39.12, pp. 4023–4033.

Liao, Haofu, Addisu Mesfin and Jiebo Luo (2018). 'Joint Vertebrae Identification and Localization in Spinal CT Images by Combining Short-and Long-Range Contextual Information'. In: *IEEE transactions on medical imaging* 37.5, pp. 1266–1275.

Lienhart, Rainer and Jochen Maydt (2002). 'An extended set of haar-like features for rapid object detection'. In: *Proceedings. international conference on image processing*. Vol. 1. IEEE, pp. I–I.

Lipton, Zachary C (2018). 'The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery.' In: *Queue* 16.3, pp. 31–57.

Listgarten, Jennifer et al. (2007). 'Difference detection in LC-MS data for protein biomarker discovery'. In: *Bioinformatics* 23.2, e198–e204.

Liu, Qin et al. (2020a). 'Addressing the batch effect issue for LC/MS metabolomics data in data preprocessing'. In: *Scientific reports* 10.1, pp. 1–13.

Liu, Qingzhong et al. (2009). 'Comparison of feature selection and classification for MALDI-MS data'. In: *BMC genomics* 10.1, S3.

Liu, Shixiang et al. (2020b). 'Recent advances on protein separation and purification methods'. In: *Advances in Colloid and Interface Science* 284, p. 102254.

Long, Jonathan, Evan Shelhamer and Trevor Darrell (2015). 'Fully convolutional networks for semantic segmentation'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Lorenzo, Pablo Ribalta et al. (2019). 'Segmenting brain tumors from FLAIR MRI using fully convolutional neural networks'. In: *Computer methods and programs in biomedicine* 176, pp. 135–148.

Lowe, David G (2004). 'Distinctive image features from scale-invariant keypoints'. In: *International journal of computer vision* 60.2, pp. 91–110.

Lundberg, Scott M and Su-In Lee (2017). 'A unified approach to interpreting model predictions'. In: *Advances in neural information processing systems*, pp. 4765–4774.

Lundervold, Alexander S and A Lundervold (2019). 'An overview of deep learning in medical imaging focusing on MRI'. In: *Zeitschrift für Medizinische Physik* 29.2, pp. 102–127.

Ma, Chunwei et al. (2018a). 'Improved peptide retention time prediction in liquid chromatography through deep learning'. In: *Analytical chemistry* 90.18, pp. 10881–10888.

Ma, Jianzhu et al. (2018b). 'Using deep learning to model the hierarchical structure and function of a cell'. In: *Nature methods* 15.4, pp. 290–298.

Ma, Jun et al. (2010). 'Hierarchical segmentation and identification of thoracic vertebra using learning-based edge detection and coarse-to-fine deformable model'. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 19–27.

Mac Aodha, Oisin et al. (2018). 'Teaching categories to human learners with visual explanations'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3820–3828.

Magni, Fulvio et al. (2010). 'Biomarkers discovery by peptide and protein profiling in biological fluids based on functionalized magnetic beads purification and mass spectrometry'. In: *Blood Transfusion* 8.Suppl 3, s92.

Makarov, Alexander (2000). 'Electrostatic axially harmonic orbital trapping: a high-performance technique of mass analysis'. In: *Analytical chemistry* 72.6, pp. 1156–1162.

Mamoshina, Polina et al. (2016). 'Applications of deep learning in biomedicine'. In: *Molecular pharmaceutics* 13.5, pp. 1445–1454.

Mamyrin, BA (2001). 'Time-of-flight mass spectrometry (concepts, achievements, and prospects)'. In: *International Journal of Mass Spectrometry* 206.3, pp. 251–266.

Maqsood, Muazzam et al. (2019). 'Transfer learning assisted classification and detection of Alzheimer's disease stages using 3D MRI scans'. In: *Sensors* 19.11, p. 2645.

Marrugal, Ángela et al. (2016). 'Proteomic-based approaches for the study of cytokines in lung cancer'. In: *Disease markers* 2016.

Mazo, Claudia et al. (2020). 'Clinical decision support systems in breast cancer: A systematic review'. In: *Cancers* 12.2, p. 369.

McCulloch, WS and W Pitts (1943). 'A logical calculus of ideas immanent in nervous activity'. In: *Bulletin of Mathematical Biophysics* 5, pp. 115–133.

McEvoy, Dustin et al. (2018). 'Enhancing problem list documentation in electronic health records using two methods: the example of prior splenectomy'. In: *BMJ quality & safety* 27.1, pp. 40–47.

Meng, Chen et al. (2016). 'Dimension reduction techniques for the integrative analysis of multi-omics data'. In: *Briefings in bioinformatics* 17.4, pp. 628–641.

Middleton, B, DF Sittig and A Wright (2016). 'Clinical decision support: a 25 year retrospective and a 25 year vision'. In: *Yearbook of medical informatics* 25.S 01, S103–S116.

Mikołajczyk, Agnieszka and Michał Grochowski (2018). 'Data augmentation for improving deep learning in image classification problem'. In: *2018 international interdisciplinary PhD workshop (IIPhDW)*. IEEE, pp. 117–122.

Milletari, Fausto, Nassir Navab and Seyed-Ahmad Ahmadi (2016). 'V-net: Fully convolutional neural networks for volumetric medical image segmentation'. In: *3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, pp. 565–571.

Miotto, Riccardo et al. (2018). 'Deep learning for healthcare: review, opportunities and challenges'. In: *Briefings in bioinformatics* 19.6, pp. 1236–1246.

Montavon, Grégoire, Wojciech Samek and Klaus-Robert Müller (2018). 'Methods for interpreting and understanding deep neural networks'. In: *Digital Signal Processing* 73, pp. 1–15.

Montavon, Grégoire et al. (2019). 'Layer-wise relevance propagation: an overview'. In: *Explainable AI: interpreting, explaining and visualizing deep learning*. Springer, pp. 193–209.

Mu, Jesse and Jacob Andreas (2020). 'Compositional explanations of neurons'. In: *arXiv preprint arXiv:2006.14032*.

Nair, Vinod and Geoffrey E Hinton (2010). 'Rectified linear units improve restricted boltzmann machines'. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814.

Nakkiran, Preetum et al. (2019). 'Deep double descent: Where bigger models and more data hurt'. In: *arXiv preprint arXiv:1912.02292*.

Nalepa, Jakub, Michal Marcinkiewicz and Michal Kawulok (2019). 'Data augmentation for brain-tumor segmentation: a review'. In: *Frontiers in computational neuroscience* 13, p. 83.

Nejati, H. et al. (2016). 'Smartphone and Mobile Image Processing for Assisted Living: Health-monitoring apps powered by advanced mobile imaging algorithms'. In: *IEEE Signal Processing Magazine* 33.4, pp. 30–48.

Nejati, Hossein et al. (2016). 'Smartphone and mobile image processing for assisted living: Health-monitoring apps powered by advanced mobile imaging algorithms'. In: *IEEE Signal Processing Magazine* 33.4, pp. 30–48.

Nesterov, Yurii (1983). 'A method for unconstrained convex minimization problem with the rate of convergence O (1/kˆ 2)'. In: *Doklady an ussr*. Vol. 269, pp. 543–547.

Noble, J Alison and Djamal Boukerroui (2006). 'Ultrasound image segmentation: a survey'. In: *IEEE Transactions on medical imaging* 25.8, pp. 987–1010.

Noble, William S (2006). 'What is a support vector machine?' In: *Nature biotechnology* 24.12, pp. 1565–1567.

Oktay, Ayse Betul and Yusuf Sinan Akgul (2011). 'Localization of the lumbar discs using machine learning and exact probabilistic inference'. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 158–165.

Olsen, Jesper V et al. (2007). 'Higher-energy C-trap dissociation for peptide modification analysis'. In: *Nature methods* 4.9, pp. 709–712.

Osheroff, Jerome A et al. (2012). *Improving outcomes with clinical decision support: an implementer's guide*. Himss Publishing.

Palagi, Patricia M et al. (2005). 'MSight: An image analysis software for liquid chromatography-mass spectrometry'. In: *Proteomics* 5.9, pp. 2381–2384.

Panesar, Arjun (2019). *Machine learning and AI for healthcare*. Springer.

Pedregosa, F. et al. (2011). 'Scikit-learn: Machine Learning in Python'. In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Peiffer-Smadja, Nathan et al. (2020). 'Machine learning for clinical decision support in infectious diseases: a narrative review of current applications'. In: *Clinical Microbiology and Infection* 26.5, pp. 584–595.

Peris-Lopez, Pedro et al. (2011). 'A comprehensive RFID solution to enhance inpatient medication safety'. In: *International journal of medical informatics* 80.1, pp. 13–24.

Petegrosso, Raphael et al. (2017). 'Transfer learning across ontologies for phenome–genome association prediction'. In: *Bioinformatics* 33.4, pp. 529–536.

Piccialli, Francesco et al. (2021). 'A survey on deep learning in medicine: Why, how and when?' In: *Information Fusion* 66, pp. 111–137.

Plewes, Donald B and Walter Kucharczyk (2012). 'Physics of MRI: a primer'. In: *Journal of magnetic resonance imaging* 35.5, pp. 1038–1054.

Pluskal, Tomáš et al. (2010). 'MZmine 2: modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data'. In: *BMC bioinformatics* 11.1, p. 395.

Podwojski, Katharina et al. (2009). 'Retention time alignment algorithms for LC/MS data must consider non-linear shifts'. In: *Bioinformatics* 25.6, pp. 758–764.

Porter, Teresita M and Mehrdad Hajibabaei (2018). 'Scaling up: A guide to high-throughput genomic approaches for biodiversity analysis'. In: *Molecular ecology* 27.2, pp. 313–338.

Qi, Da et al. (2012). 'A software toolkit and interface for performing stable isotope labeling and top3 quantification using Progenesis LC-MS'. In: *Omics: a journal of integrative biology* 16.9, pp. 489–495.

Qi, Zhongang, Saeed Khorram and Fuxin Li (2019). 'Visualizing Deep Networks by Optimizing with Integrated Gradients.' In: *CVPR Workshops*. Vol. 2.

Qin, Chunli et al. (2020). 'Residual block-based multi-label classification and localization network with integral regression for vertebrae labeling'. In: *arXiv preprint arXiv:2001.00170*.

Rashid, Mamunur et al. (2020). 'Current status, challenges, and possible solutions of EEG-based brain-computer interface: a comprehensive review'. In: *Frontiers in neurorobotics* 14, p. 25.

Rawson, Timothy M et al. (2019). 'Artificial intelligence can improve decision-making in infection management'. In: *Nature human behaviour* 3.6, pp. 543–545.

Redmon, Joseph and Ali Farhadi (2017). 'YOLO9000: better, faster, stronger'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271.

— (2018). 'Yolov3: An incremental improvement'. In: *arXiv preprint arXiv:1804.02767*.

Redmon, Joseph et al. (2016). 'You only look once: Unified, real-time object detection'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

Ren, Shaoqing et al. (2015). 'Faster r-cnn: Towards real-time object detection with region proposal networks'. In: *Advances in neural information processing systems*, pp. 91–99.

— (2016). 'Faster R-CNN: towards real-time object detection with region proposal networks'. In: *IEEE transactions on pattern analysis and machine intelligence* 39.6, pp. 1137–1149.

Ribeiro, Marco Tulio, Sameer Singh and Carlos Guestrin (2016). '" Why should I trust you?" Explaining the predictions of any classifier'. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144.

— (2018). 'Anchors: High-precision model-agnostic explanations'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1.

Rieger, Laura et al. (2020). 'Interpretations are useful: penalizing explanations to align neural networks with prior knowledge'. In: *International Conference on Machine Learning*. PMLR, pp. 8116–8126.

Ristevski, Blagoj and Ming Chen (2018). 'Big data analytics in medicine and healthcare'. In: *Journal of integrative bioinformatics* 15.3.

Ronneberger, Olaf, Philipp Fischer and Thomas Brox (2015). 'U-net: Convolutional networks for biomedical image segmentation'. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241.

Rosenblatt, F (1958). 'The perceptron: A probabilistic model for information storage and organization in the brain.' In: *Psychological Review* 65, pp. 386–408.

Röst, Hannes L et al. (2016). 'OpenMS: a flexible open-source software platform for mass spectrometry data analysis'. In: *Nature methods* 13.9, pp. 741–748.

Ruder, Sebastian (2017). 'An overview of multi-task learning in deep neural networks'. In: *arXiv preprint arXiv:1706.05098*.

Russakovsky, Olga et al. (2015). 'ImageNet Large Scale Visual Recognition Challenge'. In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252.

Samek, Wojciech et al. (2020). 'Toward Interpretable Machine Learning: Transparent Deep Neural Networks and Beyond'. In: *arXiv preprint arXiv:2003.07631*.

Sayres, Rory et al. (2019). 'Using a deep learning algorithm and integrated gradients explanation to assist grading for diabetic retinopathy'. In: *Ophthalmology* 126.4, pp. 552–564.

Schapire, Robert E (2013). 'Explaining adaboost'. In: *Empirical inference*. Springer, pp. 37–52.

Schiffman, Courtney et al. (2019). 'Filtering procedures for untargeted LC-MS metabolomics data'. In: *BMC bioinformatics* 20.1, p. 334.

Schmidt, Stefan et al. (2007). 'Spine detection and labeling using a parts-based graphical model'. In: *Biennial International Conference on Information Processing in Medical Imaging*. Springer, pp. 122–133.

Sekuboyina, Anjany et al. (2017). 'A localisation-segmentation approach for multi-label annotation of lumbar vertebrae using deep nets'. In: *arXiv preprint arXiv: 1703.04347*.

Semmlow, John L et al. (1980). 'A fully automated system for screening xeromammograms'. In: *Computers and Biomedical Research* 13.4, pp. 350–362.

Shahsavarani, Amir Mohammad et al. (2015). 'Clinical decision support systems (CDSSs): state of the art review of literature'. In: *International Journal of Medical Reviews* 2.4, pp. 299–308.

Shapley, Lloyd S (1953). 'A value for n-person games'. In: *Contributions to the Theory of Games* 2.28, pp. 307–317.

Sharma, Aman and Rinkle Rani (2021). 'A systematic review of applications of machine learning in cancer prediction and diagnosis'. In: *Archives of Computational Methods in Engineering* 28.7, pp. 4875–4896.

Shen, Ying et al. (2018). 'An ontology-driven clinical decision support system (IDDAP) for infectious disease diagnosis and antibiotic prescription'. In: *Artificial intelligence in medicine* 86, pp. 20–32.

Shi, Feng et al. (2020). 'Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for covid-19'. In: *IEEE reviews in biomedical engineering*.

Shin, Hoo-Chang et al. (2016). 'Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning'. In: *IEEE transactions on medical imaging* 35.5, pp. 1285–1298.

Shortliffe, Edward (2012). *Computer-based medical consultations: MYCIN*. Vol. 2. Elsevier.

Shortliffe, Edward H and Bruce G Buchanan (1975). 'A model of inexact reasoning in medicine'. In: *Mathematical biosciences* 23.3-4, pp. 351–379.

Shortliffe, Edward H et al. (1975). 'Computer-based consultations in clinical therapeutics: explanation and rule acquisition capabilities of the MYCIN system'. In: *Computers and biomedical research* 8.4, pp. 303–320.

Shrestha, Ajay and Ausif Mahmood (2019). 'Review of deep learning algorithms and architectures'. In: *IEEE Access* 7, pp. 53040–53065.

Shrikumar, Avanti et al. (2016). 'Not just a black box: Learning important features through propagating activation differences'. In: *arXiv preprint arXiv:1605.01713*.

Shu, Li-Qi et al. (2019). 'Application of artificial intelligence in pediatrics: past, present and future'. In: *World Journal of Pediatrics* 15.2, pp. 105–108.

Siddique, Nahian et al. (2021). 'U-net and its variants for medical image segmentation: A review of theory and applications'. In: *IEEE Access*.

Silveira, Margarida et al. (2009). 'Comparison of segmentation methods for melanoma diagnosis in dermoscopy images'. In: *IEEE Journal of Selected Topics in Signal Processing* 3.1, pp. 35–45.

Sim, Ida et al. (2001). 'Clinical decision support systems for the practice of evidence-based medicine'. In: *Journal of the American Medical Informatics Association* 8.6, pp. 527–534.

Simard, Patrice Y, David Steinkraus, John C Platt et al. (2003). 'Best practices for convolutional neural networks applied to visual document analysis.' In: *Icdar*. Vol. 3. 2003. Citeseer.

Simonyan, Karen, Andrea Vedaldi and Andrew Zisserman (2013a). 'Deep inside convolutional networks: Visualising image classification models and saliency maps'. In: *arXiv preprint arXiv:1312.6034*.

— (2013b). 'Deep inside convolutional networks: Visualising image classification models and saliency maps'. In: *arXiv preprint arXiv:1312.6034*.

Simonyan, Karen and Andrew Zisserman (2014). 'Very deep convolutional networks for large-scale image recognition'. In: *arXiv preprint arXiv:1409.1556*.

Singhal, Neelja et al. (2015). 'MALDI-TOF mass spectrometry: an emerging technology for microbial identification and diagnosis'. In: *Frontiers in microbiology* 6, p. 791.

Smilkov, Daniel et al. (2017). 'Smoothgrad: removing noise by adding noise'. In: *arXiv preprint arXiv:1706.03825*.

Smith, Colin A et al. (2006). 'XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification'. In: *Analytical chemistry* 78.3, pp. 779–787.

Smoller, Jordan W (2018). 'The use of electronic health records for psychiatric phenotyping and genomics'. In: *American Journal of Medical Genetics Part B: Neuropsychiatric Genetics* 177.7, pp. 601–612.

Souza, Gustavo Henrique Martins Ferreira, Paul C Guest and Daniel Martins-de Souza (2017). 'LC-MS E, multiplex MS/MS, ion mobility, and label-free quantitation in clinical proteomics'. In: *Multiplex Biomarker Techniques*. Springer, pp. 57–73.

Springenberg, Jost Tobias et al. (2014). 'Striving for simplicity: The all convolutional net'. In: *arXiv preprint arXiv:1412.6806*.

Srivastava, Nitish et al. (2014). 'Dropout: a simple way to prevent neural networks from overfitting'. In: *The journal of machine learning research* 15.1, pp. 1929–1958.

Sun, Chen et al. (2017). 'Revisiting unreasonable effectiveness of data in deep learning era'. In: *Proceedings of the IEEE international conference on computer vision*, pp. 843–852.

Sundararajan, Mukund, Ankur Taly and Qiqi Yan (2017). 'Axiomatic attribution for deep networks'. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 3319–3328.

Sutskever, Ilya, Oriol Vinyals and Quoc V Le (2014). 'Sequence to sequence learning with neural networks'. In: *Advances in neural information processing systems*, pp. 3104–3112.

Sutton, Reed T et al. (2020). 'An overview of clinical decision support systems: benefits, risks, and strategies for success'. In: *NPJ digital medicine* 3.1, pp. 1–10.

Sutton, Richard (1986). 'Two problems with back propagation and other steepest descent learning procedures for networks'. In: *Proceedings of the Eighth Annual Conference of the Cognitive Science Society, 1986*, pp. 823–832.

Suzani, Amin et al. (2015). 'Fast automatic vertebrae detection and localization in pathological ct scans-a deep learning approach'. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 678–686.

Syka, John EP et al. (2004). 'Peptide and protein sequence analysis by electron transfer dissociation mass spectrometry'. In: *Proceedings of the National Academy of Sciences* 101.26, pp. 9528–9533.

Szegedy, Christian et al. (2015). 'Going deeper with convolutions'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.

Szegedy, Christian et al. (June 2016). 'Rethinking the Inception Architecture for Computer Vision'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tan, Chee Fai et al. (2016). 'The application of expert system: A review of research and applications'. In: *ARPN Journal of Engineering and Applied Sciences* 11.4, pp. 2448–2453.

Tan, Chuanqi et al. (2018). 'A survey on deep transfer learning'. In: *International conference on artificial neural networks*. Springer, pp. 270–279.

Tang, Xinyao et al. (2018). 'On combining active and transfer learning for medical data classification'. In: *IET Computer Vision* 13.2, pp. 194–205.

Taylor, J Alex and Richard S Johnson (1997). 'Sequence database searches via de novo peptide sequencing by tandem mass spectrometry'. In: *Rapid communications in mass spectrometry* 11.9, pp. 1067–1075.

Thomas, Armin W et al. (2019). 'Analyzing neuroimaging data through recurrent deep learning models'. In: *Frontiers in neuroscience* 13, p. 1321.

Tiwari, Arti, Shilpa Srivastava and Millie Pant (2020). 'Brain tumor segmentation and classification from magnetic resonance images: Review of selected methods from 2014 to 2019'. In: *Pattern Recognition Letters* 131, pp. 244–260.

Toh, Tzen S, Frank Dondelinger and Dennis Wang (2019). 'Looking beyond the hype: applied AI and machine learning in translational medicine'. In: *EBioMedicine* 47, pp. 607–615.

Topol, Eric J (2019). 'High-performance medicine: the convergence of human and artificial intelligence'. In: *Nature medicine* 25.1, pp. 44–56.

Toro, Carlos et al. (2012). 'Using set of experience knowledge structure to extend a rule set of clinical decision support system for alzheimer's disease diagnosis'. In: *Cybernetics and Systems* 43.2, pp. 81–95.

Toshev, Alexander and Christian Szegedy (2014). 'Deeppose: Human pose estimation via deep neural networks'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1653–1660.

Tran, Ngoc Hieu et al. (2017). 'De novo peptide sequencing by deep learning'. In: *Proceedings of the National Academy of Sciences* 114.31, pp. 8247–8252.

Tran, Ngoc Hieu et al. (2019). 'Deep learning enables de novo peptide sequencing from data-independent-acquisition mass spectrometry'. In: *Nature methods* 16.1, pp. 63–66.

Trebeschi, Stefano et al. (2017). 'Deep learning for fully-automated localization and segmentation of rectal cancer on multiparametric MR'. In: *Scientific reports* 7.1, pp. 1–9.

Tu, Zhuowen et al. (2008). 'Brain anatomical structure segmentation by hybrid discriminative/generative models'. In: *IEEE transactions on medical imaging* 27.4, pp. 495–508.

Tuli, Leepika et al. (2012). 'Using a spike-in experiment to evaluate analysis of LC-MS data'. In: *Proteome science* 10.1, p. 13.

Uijlings, Jasper RR et al. (2013). 'Selective search for object recognition'. In: *International journal of computer vision* 104.2, pp. 154–171.

Välikangas, Tommi, Tomi Suomi and Laura L Elo (2018). 'A systematic evaluation of normalization methods in quantitative label-free proteomics'. In: *Briefings in bioinformatics* 19.1, pp. 1–11.

Viceconti, Marco, Peter Hunter and Rod Hose (2015). 'Big data, big knowledge: big data for personalized healthcare'. In: *IEEE journal of biomedical and health informatics* 19.4, pp. 1209–1215.

Villanueva, Josep et al. (2006). 'Serum peptidome patterns that distinguish metastatic thyroid carcinoma from cancer-free controls are unbiased by gender and age'. In: *Molecular & Cellular Proteomics* 5.10, pp. 1840–1852.

Wadhwa, Vaibhav et al. (2020). 'Physician sentiment toward artificial intelligence (AI) in colonoscopic practice: a survey of US gastroenterologists'. In: *Endoscopy international open* 8.10, E1379–E1384.

Wang, Fakai et al. (2021). 'Automatic vertebra localization and identification in ct by spine rectification and anatomically-constrained optimization'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5280–5288.

Wang, Hui et al. (2016). 'The clinical impact of recent advances in LC-MS for cancer biomarker discovery and verification'. In: *Expert review of proteomics* 13.1, pp. 99–114.

Watson, Robert E (2015). 'Lessons learned from MRI safety events'. In: *Current Radiology Reports* 3.10, pp. 1–7.

Wells, J Mitchell and Scott A McLuckey (2005). 'Collision-induced dissociation (CID) of peptides and proteins'. In: *Methods in enzymology* 402, pp. 148–185.

Werbos, P (1974a). 'Beyond regression: new tools for prediction and analysis in the behavioral science.' In: *PhD diss*, pp. 65–78.

Werbos, Paul (1974b). 'Beyond regression:" new tools for prediction and analysis in the behavioral sciences'. In: *Ph. D. dissertation, Harvard University.*

Wheeler, David L et al. (2007). 'Database resources of the national center for biotechnology information'. In: *Nucleic acids research* 36.suppl_1, pp. D13–D21.

Wilson, Fiona et al. (2021). 'Prevalence and risk factors for back pain in sports: a systematic review with meta-analysis'. In: *British Journal of Sports Medicine* 55.11, pp. 601–607.

Wold, Svante, Kim Esbensen and Paul Geladi (1987). 'Principal component analysis'. In: *Chemometrics and intelligent laboratory systems* 2.1-3, pp. 37–52.

Wolf, Thomas et al. (2020). 'Transformers: State-of-the-art natural language processing'. In: *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45.

Xu, Qian, Evan Wei Xiang and Qiang Yang (2010). 'Protein-protein interaction prediction via collective matrix factorization'. In: *2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, pp. 62–67.

Yadav, Garima, Saurabh Maheshwari and Anjali Agarwal (2014). 'Contrast limited adaptive histogram equalization based enhancement for real time video system'. In: *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on*. IEEE, pp. 2392–2397.

Yi, Xin, Ekta Walia and Paul Babyn (2019). 'Generative adversarial network in medical imaging: A review'. In: *Medical image analysis* 58, p. 101552.

Yi-de, Ma, Liu Qing and Qian Zhi-Bai (2004). 'Automated image segmentation using improved PCNN model based on cross-entropy'. In: *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.* IEEE, pp. 743–746.

Yosinski, Jason et al. (2014). 'How transferable are features in deep neural networks?' In: *Advances in neural information processing systems*, pp. 3320–3328.

Young, Tom et al. (2018). 'Recent trends in deep learning based natural language processing'. In: *ieee Computational intelligenCe magazine* 13.3, pp. 55–75.

Yu, Sheng et al. (2015). 'Toward high-throughput phenotyping: unbiased automated feature extraction and selection from knowledge sources'. In: *Journal of the American Medical Informatics Association* 22.5, pp. 993–1000.

Zeiler, Matthew D (2012). 'Adadelta: an adaptive learning rate method'. In: *arXiv preprint arXiv:1212.5701*.

Zeiler, Matthew D and Rob Fergus (2014). 'Visualizing and understanding convolutional networks'. In: *European conference on computer vision*. Springer, pp. 818–833.

Zhan, Yiqiang et al. (2015). 'Cross-modality vertebrae localization and labeling using learning-based approaches'. In: *Spinal Imaging and Image Analysis*. Springer, pp. 301–322.

Zhang, Yu et al. (2021). 'A survey on neural network interpretability'. In: *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Zhao, Nanxuan et al. (2021). 'What Makes Instance Discrimination Good for Transfer Learning?' In: *International Conference on Learning Representations*.

Zhou, Bolei et al. (2017a). 'Places: A 10 million image database for scene recognition'. In: *IEEE transactions on pattern analysis and machine intelligence* 40.6, pp. 1452–1464.

Zhou, Jian and Olga G Troyanskaya (2015). 'Predicting effects of noncoding variants with deep learning–based sequence model'. In: *Nature methods* 12.10, pp. 931–934.

Zhou, Xie-Xuan et al. (2017b). 'pDeep: predicting MS/MS spectra of peptides with deep learning'. In: *Analytical chemistry* 89.23, pp. 12690–12697.

Zhou, Yi-Tong and Rama Chellappa (1988). 'Computation of optical flow using a neural network.' In: *ICNN*, pp. 71–78.

Zintgraf, Luisa M et al. (2017). 'Visualizing deep neural network decisions: Prediction difference analysis'. In: *arXiv preprint arXiv:1702.04595*.

Zitnik, Marinka, Monica Agrawal and Jure Leskovec (2018). 'Modeling polypharmacy side effects with graph convolutional networks'. In: *Bioinformatics* 34.13, pp. i457–i466.

Zohora, Fatema Tuz et al. (2019). 'DeepIso: A Deep Learning Model for Peptide Feature Detection from LC-MS map'. In: *Scientific reports* 9.1, pp. 1–13.

Zou, Hui and Trevor Hastie (2005). 'Regularization and variable selection via the elastic net'. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2, pp. 301–320.