Machine Learning and Image Processing for Clinical Outcome Prediction: Applications in Medical Data from Patients with Traumatic Brain Injury, Ulcerative Colitis, and Heart Failure

by

Heming Yao

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Bioinformatics)
in The University of Michigan
2021

Doctoral Committee:

Professor Kayvan Najarian, Chair
Associate Professor Alan Boyle
Professor Harm Derksen
Associate Professor Alla Karnovsky
Professor Gilbert S. Omenn
Associate Professor Ryan W. Stidham
Associate Professor Craig A. Williamson

Heming Yao

hemingy@umich.edu

ORCID iD: 0000-0001-9020-5330

To my parents, Yurong Zhang and Yunlin Yao

ACKNOWLEDGEMENTS

This dissertation would not have been possible without many individuals. I would like to express my gratitude to all people that have helped, encouraged, and supported me during my graduate study.

First and foremost, I would like to thank my advisor, Dr. Kayvan Najarian, for his mentorship. He is a brilliant scientist who has profound and broad knowledge in the machine learning research field. I am grateful for the opportunity of working on a research project in his class at the beginning of my graduate study and joining his lab after my lab rotation. Whenever I encountered difficulties with my research, Dr. Najarian was always there, standing by me, listening and answering my questions. He has a super active and creative mind. His insight expanded my horizon and deepened my understanding of the research world; his enthusiasm and excitement for research in healthcare inspired my passion for solving the challenges of practical problems. I learn a lot from him and will continue to benefit from these characteristics.

I would also like to thank Dr. Ryan Stidham, the clinical lead of my research project in colonoscopy video analysis. This existing project is not possible without his guidance and efforts. I am grateful for him being incredibly generous with his time and advice every time I sought him out. His feedback and suggestion immensely helped me improve my skills and research work. It's been my absolute pleasure to work with Dr. Stidham.

My dissertation committee members were all invaluable mentors. Dr. Harm Derksen, an expert in mathematics, proposed brilliant ideas to solve the optimization issue

with my algorithm. I thank Dr. Gil Omenn, Dr. Alan Boyle, Dr. Alla Karnovsky for their insight and feedback on my research work. I want to thank Dr. Craig Willianmson for his support on my first graduate research project on hematoma segmentation. It is the first project that inspires my interest and motivation in developing machine learning algorithms for medical applications. Dr. Crag Willianmson's invaluable advice and help sharpened my skills and given me confidence.

I would like to express my appreciation to the Najarian Lab and the Department of Computational Medicine & Bioinformatics for their support and encouragement. In particular, I would like to thank Dr. Jonathan Gryak for his countless support to projects in the lab and his valuable advice to my research and professional development. I would also like to thank Julia Eussen and Dr. Margit Burmeister for helping me navigate through the academic environment in the department.

I am so thankful for my Ph.D. cohort, labmates, colleagues, and friends I've met in and out of the University of Michigan. Together with them, I have so many enjoyable experiences and memories.

Furthermore, I would like to thank my significant other and biggest supporter, Niankai. His company from the start of my Ph.D. study is incredibly valuable. His love and encouragement helped me get through the most difficult periods of graduate study and the pandemic. I would like to thank my adorable cats Gracie and Picabo, who are the most cheerful and energetic creatures at home. They are lovely and warm. During the last two years with the pandemic, they came to me and have brought endless joy to my life.

Finally, I would like to express my deepest thanks and love to my parents. Their endless love and support encourage me always being positive towards any challenges in my life. They are the ones who help me become who I am.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

LIST OF APPENDICES

Appendix

LIST OF ABBREVIATIONS

ACM Active Contour Model

AI Artificial Intelligence

ANFIS Adaptive Network-based Fuzzy Inference System

ATE Absolute trajectory error

AUC Area Under the Receiver Operating Characteristic Curve

AUPRC Area Under the Precision-Recall Curve

CDS Clinical Decision Support

CNN Convolutional Neural Network

CPE Corrected Photometric Error

CT Computed Tomography

DISTWLK Total Distance Walked in 5 Minutes

DL Deep Learning

EBM Explainable Boosting Machine

EDH Epidural Hematoma

EF Left Ventricle Ejection Fraction Severity Score

EHR Electronic Health Record

FOE focus of expansion

FDA U.S. Food & Drug Adminstration

FPS Frames Per Second

GCS Glasgow Coma Scale

GDMT Guideline Directed Medical Therapy

GFR Glomerular Filtration Rate

GLCM Gray-level Co-occurrence Matrix

HF Heart Failure

HGB Hemoglobin

HT Heart Transplantation

IBD Inflammatory Bowel Diseases

ICC Intraclass Correlation Coefficient

INTERMACS Interagency Registry for Mechanically Assisted Circulatory Support

IPH Intraparenchymal Hematoma

IVH Intraventricular Hematoma

LVAD Left Ventricle Assist Device

LVDEM Left Ventricular Dimension in Diastole

LYMPH Lymphocyte Percentage

MCS Mechanical Circulatory Support

MES Mayo Endoscopic Subscore

MITRGRG Mitral Regurgitation

ML Machine Learning

NYHA New York Heart Association

PHQ-8 Eight-item Patient Health Questionnaire Depression Scale

PROTECT Traumatic Brain Injury, Experimental Clinical Treatment

pVO2 Peak Oxygen Consumption During a Maximal Cardiopulmonary Exercise Test

REVIVAL Registry Evaluation of Vital Information for VADs in Ambulatory Life

ROI Region of Interest

RPE Relative Pose Error

SD Standard Deviation

SDH Subdural Hematoma

SLIC Simple Linear Iterative Clustering

SVM Support Vector Machines

SYSBP Systolic Blood Pressure

TBI Traumatic Brain Injury

TCH Total Cholesterol

TS Takagi-Sugeno

UC Ulcerative Colitis

GTSPDTM Gait Speed During a 15 Feet Walk Test

ABSTRACT

Artificial intelligence (AI) and machine learning (ML) have achieved extensive success in many fields. They are powerful in pattern recognition and function modeling. The digitization of health data provides an important opportunity for improving care delivery and patient management through the AI-based clinical decision-support (CDS) system. Medical images are important components in evaluating the disease severity. While the human's interpretation of medical images is subjective and qualitative, AI-based models can analyze those data in a more reproducible, quantitative, and less expensive way. With clinical observations and quantitative findings extracted from medical images, ML methods can be used to learn and discover knowledge. The automated CDS system can provide recommendations on diagnosis, treatment, and outcome prediction by leveraging massive medical data. Those systems can facilitate drug development, disease pathology research, and clinical practice.

This dissertation investigates medical image analysis and CDS systems development in a more reliable, interpretable manner. Limitations exist in applying AI/ML techniques in medical problems. Medical data may have high variability in terms of the patient population, collection site, equipment, and imaging protocols. It is crucial that the ML and image processing algorithms have a good generalizability and can be reliably applied to unseen patient data. In addition, a broad spectrum of AI/ML methods is among the "black box" models. The lack of justification leads to concerns and hesitations of using AI/ML techniques in clinical or research practice. Features

with clinical meaning and models that can be well explained can gain more trust and are more favorable to end-users.

In this dissertation, several AI-based CDS systems have been designed and implemented to facilitate clinical and research practice. Novel algorithms are proposed to overcome the challenges of applying AI/ML techniques. To improve the generalizability of the deep learning models, a robust learning algorithm is proposed to encourage the network to be invariant to hematoma intensity variability. A Scale Module and filter pruning technique are proposed to reduce the network's size and complexity. To improve the interpretability of the CDS systems, a transparent ML algorithm is proposed based on tropical geometry and fuzzy logic, which can learn humanly understandable rules from the dataset and integrate existing domain knowledge to facilitate the model training. Domain knowledge plays an important role in the design of CDS systems. With automated image analysis methods, quantitative and objective measurements are extracted to capture the patient's condition and disease characteristics in a meaningful and reproducible way. The proposed CDS systems have been validated using data collected from routine practice and clinical trials. The datasets used in this dissertation are from multiple medial centers, which increases the generalizability of the proposed frameworks and trained models.

This work aims to research the capacity of AI models toward fully automated CDS systems that can replicate expert judgment and provide insight for the patient. Efforts have been made to improve the generalizability and interpretability of AI/ML models, which are the major limitations that hinder a broad application of AI techniques in practice. The proposed algorithms and strategies in this dissertation leverage big data to improve the healthcare system and disease research. Additionally, the proposed methods are transferable beyond the target application. The contributions of this dissertation have a meaningful impact on applying AI-based systems to clinical and research practice.

CHAPTER I

Introduction

1.1 Background and Motivation

The massive influx of medical data from medical imaging storage, Electronic Medical Records (EHR), and clinical trials have aroused increasing enthusiasm for data-driven-based applications [1, 2]. A large amount of data are valuable to study the diagnosis, treatment, and pathology of diseases. While the human's analysis and interpretation of those data are tedious, subjective, and error-prone [2, 3, 4], the development and extensive success of machine learning (ML) and deep learning (DL) techniques across many fields show the capacity of artificial intelligence (AI) in pattern recognition and knowledge learning.

The digitization of health data provides an important opportunity for improved care delivery and patient management through the AI-based clinical decision support (CDS) system. Figure 1.1 shows an overview of the building of CDS systems with AI techniques, which will be investigated in this dissertation. The knowledge base includes patient data collected from routine examinations and hospitalizations. With image processing and DL models, anatomical structures and disease-related abnormalities can be detected and segmented from medical images and videos. With detection and segmentation results, quantitative measurements can be calculated to describe the patient's condition. Processing medical images and videos in an auto-

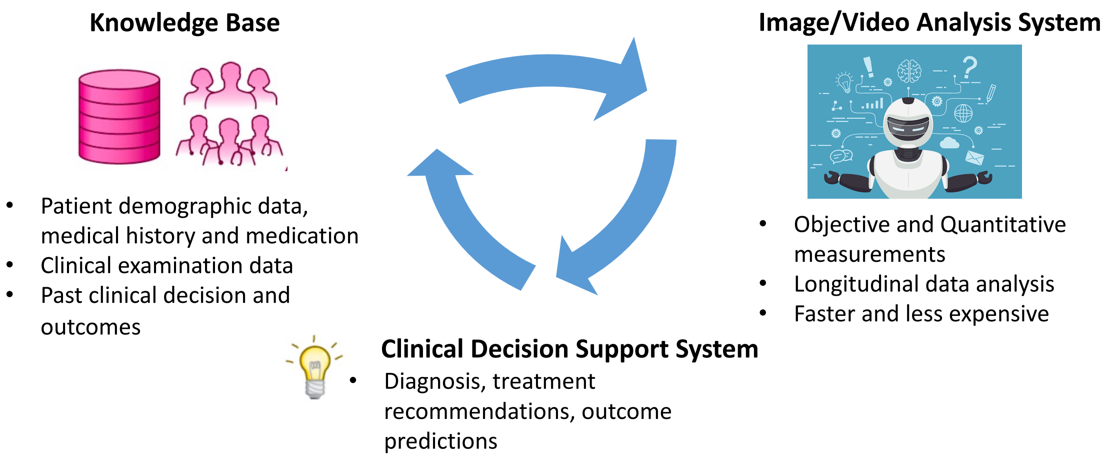

Knowledge Base
• Patient demographic data, medical history and medication
• Clinical examination data
• Past clinical decision and outcomes
Image/Video Analysis System
• Objective and Quantitative measurements
• Longitudinal data analysis
• Faster and less expensive
Clinical Decision Support System
• Diagnosis, treatment recommendations, outcome predictions

Figure 1.1: An overview of building a CDS system with AI.

mated way can decrease medical costs by reducing work and providing more objective and reproducible outputs. Based on the calculated quantitative measurements from medical images and videos and tabular features from EHR, ML models will help build diagnostic and prognostic models. Those models provide clinicians with recommendations of diagnosis, treatment, and outcome perspective. With the assistant of the AI-based CDS system, clinicians can leverage information from large amounts of past patient data more effectively and make a final decision for individual patients. After that, the decision made by clinicians can be added back to the knowledge base to iteratively optimize the developed AI-based image/video analysis methods and CDS systems.

AI-based systems are promising in improving healthcare delivery and patient management. However, limitations exist, which have created a serious concern and hesitation over using AI techniques in clinical practice, where reliability and interpretability play a vital role. The first concern is the lack of generalizability. ML and DL models are powerful in extracting patterns from the data and modeling the relationship between the extracted patterns and the targets. However, the pattern extraction learned from one data cohort may not work on other data cohorts [2]. Previous studies [5, 6, 7, 8] have shown that many trained ML and DL models are vulnerable to adversarial attacks. Even trivial noises and data transformations that are negligible to human eyes can significantly change the model's output. Additionally, in healthcare applications, data may be collected from multiple countries and medical centers, where the medical devices are from various manufacturers, and imaging protocols have different standards. A poor generalizability makes the trained model being less applicable to data underrepresented in the training dataset. The lack of interpretability is another primary concern [9, 10]. Decision-makers in high-stakes fields, such as medicine, are much less likely to trust recommendations for which no clear justification is provided. However, many AI/ML methods are "black box" models, where high non-linear functions are approximated in pattern recognition. It isn't straightforward for humans to interpret those functions [11]. The missing of justification also poses challenges in the model's diagnosis. It is challenging to evaluate or estimate the generalizability of a trained model on an unseen data cohort. Moreover, if a trained model makes a wrong decision on specific groups of data, it is not explicit how to improve the model. Additional challenges include the shortage of annotated data (especially high-quality ones) and the lack of effective ways to integrate domain knowledge into AI models. More efforts are needed to overcome those challenges of applying AI techniques in medical problems.

1.2 Objectives

This dissertation aims to build reliable and interpretable AI-based CDS systems with real-world medical applications.

1.2.1 Interpreting medical data and developing CDS systems

Collaborating with clinicians, practical and critical decision-making problems will be formulated. The knowledge base will be constructed for individual medical applications, which includes patient demographic information, medications, clinical measurements, and medical images/videos from clinical examinations. In addition, adjudicated clinical decisions or patient outcomes will be collected or annotated.

Medical images and videos contain comprehensive information about the patient. I will develop automated image classification, segmentation, and regression algorithms to effectively identify the region of interest (ROI) and effectively estimate relevant parameters. Quantitative and clinically interpretable features will be designed and calculated to represent the patient's condition. Novel features representations from medical images and videos will be investigated. In our hypothesis, the feature representation from the automated image/video analysis system can better capture the patient's condition and show a higher diagnostic and prognostic value compared with conventional clinical parameters from human reviewers.

With the curated datasets and novel feature representation extracted from the medical images and videos, CDS models will be developed to provide diagnosis, treatment, and outcomes recommendations. The performance of the proposed automated CDS tools will be compared with conventional methods used in practice.

1.2.2 Improving the generalizability of ML/DL models

Several strategies will be proposed to overcome specific challenges and improve the generalizability of the ML/DL models in individual applications. Those strategies will include (1) building more reliable and robust loss functions; (2) extracting interpretable features based on domain knowledge from clinicians; (3) fusing domain knowledge into the network to facilitate the model learning knowledge from the data; (4) reducing the network size by automated filter pruning. The proposed strategies will be validated and analyzed on practical applications. Comparison between the proposed algorithms and existing techniques will be performed to demonstrate the superiority.

1.2.3 Improving the interpretability of the decision-making models

An interpretable ML algorithm will be developed based on the concept of soft computing and tropical geometry. The proposed algorithm will allow the representation of the knowledge created and stored in the model. As such, justifications for the resulting recommendations and predictions would be transparent to end-users. In addition, the proposed network will be able to incorporate approximate domain knowledge directly into the model training. In the proposed algorithm, tropical geometry will be used to formulate the differentiable operations used in the model, and an effective optimization algorithm will be designed to avoid the disadvantages of conventional soft computing paradigms such as fuzzy logic and Bayesian networks.

The classification performance of the proposed interpretable ML algorithm will be compared with established ML models. The learned rules will be extracted from the trained models and validated by clinical experts. We expect that (1) the proposed algorithm can achieve a comparable classification performance with other established ML models; (2) the trained model can correctly identify clinically important features and organize them in understandable rules; (3) the trained model may discover new rules that are reasonable but haven't been well recognized in the field.

In addition, for the proposed CDS system, extracting meaningful features based on domain knowledge can also help the interpretability of the decision-making models.

1.3 Dissertation Outline

In this dissertation, novel ML, DL, and computer vision algorithms are developed to extract feature representation from medical images and videos. With the extracted feature representation, AI-based CDS systems are designed to provide outcome prediction and treatment recommendations. Compared with manual human interpretation of medical data, the extracted features from the designed automated

AI-based system are more objective, quantitative and comprehensive. The proposed algorithms and strategies have been implemented and validated on practical applications with various data types and diseases.

Chapter II presents an automated hematoma evaluation and CDS system on Computed Tomography (CT) scans from patients with acute traumatic brain injury (TBI). In this design, a convolutional neural network (CNN) that fuses multi-scale features and a robust loss function are proposed to segment acute hematoma with a better generalizability. Quantitative volume features are calculated and combined with clinical variables to build a 6-month mortality prediction model. The proposed hematoma segmentation and outcome prediction models have been validated on CT scans from a large multicenter clinical trial. The mortality model achieves a significantly better performance than a widely-used logistic-regression model that only uses qualitative CT features from human reviewers. Feature importance analysis shows that the calculated hematoma volumes in anatomical regions contribute most to the proposed mortality prediction model.

Chapter III discusses my work on colonoscopy video analysis. Endoscopic scoring is an important component in the colon's disease severity. Mayo endoscopic score is one of the most commonly used scoring schemes for ulcerative colitis (UC) but is limited by its simplicity and subjectivity. This chapter proposes image classifications and location estimation models to recognize disease severity, relative location, and anatomical colon segment of individual frames from a colonoscopy video. Based on the context understanding, disease spatial severity distribution over the entire colon can be derived. Quantitative features such as statistics features in anatomical colons segments are extracted to build a comprehensive patient profile to characterize the patient's condition better. This feature representation has been validated on large clinical trial datasets, showing a higher diagnostic and prognostic value than the conventional severity score.

Chapter IV introduces my work on developing an interpretable ML algorithm for clinical decision-making. Many established ML algorithms are limited by their "black-box" property. In the proposed algorithm, the input variables are encoded into concepts commonly used in human logic. The input space will be divided into subspaces by a combination of concepts, and the relationship between the subspaces and target classes will be modeled. With a trained model, subspaces contributing to the target classes will be extracted and interpreted as rules. The proposed algorithm has been validated using both synthetic datasets and a heart failure (HT) dataset. From the experimental results, the proposed network can achieve comparable classification performance with other established ML algorithms and extract humanly understandable rules from data in my experiments. Moreover, existing domain knowledge can be easily formulated as rules and integrated into the proposed model to facilitate the model training.

Chapter V discusses additional work that solves the challenges of applying ML and DL techniques on practical applications. (1) A filter-pruning technique is proposed to reduce the size of a trained model and speed up the inference phase. A Scale Module is designed to estimate the importance of filters. The scale module will reduce the contributions from filters with less importance gradually. After the model training, users can eliminate redundant filters directly without hindering the performance of the model. (2) An active learning framework is proposed to overcome the challenge of the shortage of annotated data. An initial classifier is built on the initial training dataset in the proposed active learning framework and selects the most informative data samples from the unlabeled data pool. The chosen data samples will be annotated by human reviewers and added to the training set. The classifier's performance will increase with several iterations by learning from the most informative data samples. From my experiment, compared with the regular learning framework, the classifier trained from active learning achieves a comparative performance with a much smaller training set.

Chapter VI gives a conclusion to the research work in this dissertation. A discussion is given on the impact of the proposed CDS systems and the efforts in improving the generalizability and interpretability of ML/DL models. The contributions of this dissertation are highlighted, and insights on future work are provided.

CHAPTER II

Quantitative Hematoma Evaluation on CT scans and Outcome Prediction for Patients with Traumatic Brain Injury

2.1 Introduction

TBI is caused by a blow or jolt to the head that causes temporary, or permanent cerebral dysfunction [12]. As a major cause of death and disability, especially in children and young adults, TBI is a growing healthcare burden worldwide. The lifetime economic cost of TBI in the United States is approximately $76.5 billion, including direct and indirect medical costs [13].

The consequences of TBI can worsen rapidly without timely diagnosis and treatment. Prognostic models with data collected in the first 24 hours are essential to support early clinical decision-making. CT is the imaging modality of choice during the first 24 hours after brain injury, especially for unconscious patients in the emergency room, because of its low cost, fast imaging capability, and availability [14]. Acute brain hematoma detection and evaluation from CT scans are critical to both TBI diagnosis, and patient management [15]. Previous studies show that the shape and volume of brain hematoma are powerful predictors of mortality and morbidity in patients with TBI [16, 17, 18]. Brain hematoma volumes can also be used as an indicator for surgical management [19, 20]. An epidural hematoma (EDH) greater than 30 cm^3 is recommended for surgical evacuation regardless of the patient's Glasgow Coma Scale (GCS) score [20]. In addition, the evaluation of brain hematoma is important for epidemiologic studies and agent efficacy estimation [21, 22]. Hematoma within the intracranial compartment has five subtypes depending on its anatomical location: EDH, subdural hematoma (SDH), subarachnoid hematoma (SAH), intraparenchymal hematoma (IPH), and intraventricular hematoma (IVH). In this study, we will group all subtypes of hematoma together as "total hematoma".

Hematoma volumes can be estimated by manually delineating the hematoma boundary through CT scans, which is labor-intensive, taking around 20-30 minutes for an experienced radiologist [23]. In practice, the ABC/2 method is widely used by clinicians to measure hematoma volume [24]. The ABC/2 volume estimation assumes that hematomas are roughly ellipsoidal: the CT slice with the largest hematoma is first identified, after which hematoma volume is estimated. The ABC/2 method is fast but doesn't always produce a good estimate, especially in acute cases where regions of active bleeding can be thin or scattered. Previous studies have shown that the ABC/2 method overestimates hematoma volume by 10 to 40 percent [25, 26]. Therefore, an automated and accurate brain hematoma segmentation algorithm can help clinicians quantitatively estimate brain hematoma volumes and shapes, which will significantly facilitate TBI patient management and outcome prediction.

Many automated brain hematoma segmentation methods have been proposed. A number of segmentation methods start with initial hematoma masks and optimize them via level set techniques. In [27], an adaptive threshold was used to filter out hematoma candidates, after which a multi-resolution binary level set algorithm was applied to segment hematomas. A spatial fuzzy c-means clustering algorithm was proposed in [28] to initialize the region-based active contour model. These methods are sensitive to the initialized masks, requiring many iterations to achieve convergence, and are prone to becoming trapped in local minima.

In recent years, DL methods have been applied to acute brain hematoma segmentation. A combination of an autoencoder network and region-based active contour model was proposed to segment acute intracranial hematoma [29]. A 3D U-Net was implemented in [30] to take advantage of 3D contextual information. In [31], a hybrid 2D/3D mask ROI-based architecture was designed to detect the hematomas and perform segmentation. While these methods achieve good hematoma detection and segmentation, several limitations still exist. First, in acute TBI cases, the volumes and shapes of hematomas vary substantially. Existing CNN-based hematoma segmentation methods extract high-level image features with similarly sized receptive fields, which prevents the fusion of features from multiple scales. Secondly, most of the methods were only validated using CT scans from the same institution with identical imaging protocols. While a 3D neural network can help integrate more comprehensive contextual information, the model's performance may be affected by the slice spacing, which varies across CT scans from different health centers. Thirdly, the contribution of automated volume segmentation and estimation from CT scans to outcome prediction has not been fully explored.

In this chapter, a CDS system that predicts the 6-month mortality of patients with acute TBI is developed based on admission data, which supports early clinical decision-making before in-hospital therapeutic interventions. A novel Multi-view CNN with a mixed loss function is proposed to improve the performance and generalizability of acute hematoma segmentation on brain CT scans. After hematoma segmentation, the total volume of acute brain hematoma in the entire brain and in the individual anatomical regions are analyzed. After that, a ML algorithm for mortality classification is trained using a combination of clinical observations and features extracted from the automated hematoma segmentation. The main contributions of the work in this chapter are as follows:

1. A novel Multi-view CNN architecture with dilated convolution is proposed, where features from different scales are extracted and fused to improve the segmentation performance. From Multi-view CNN, the features for generating segmentation masks and features for hematoma identification are decoupled. Our results show that the network can lead to both a finer hematoma segmentation and better hematoma identification accuracy.

2. A novel mixed loss function calculated using CT scans whose contrasts are adjusted by different window centers and widths is utilized to improve the generalizability of the network. From the experimental results, the proposed loss function reduces the network's sensitivity to noise and subtle changes in appearance. The proposed algorithm is named as "robust learning".

3. The proposed hematoma segmentation framework has been trained and tested using CT scans from multiple institutions with different acquisition and imaging protocols. The proposed Multi-view CNN with mixed loss achieves an average Dice coefficient of 0.675 from 5-fold cross-validation and 0.697 on an independent test set. For volume estimation, the intraclass correlation coefficient (ICC) between the estimated hematoma volumes and annotated hematoma volumes is 0.959 from 5-fold cross-validation and 0.966 on the independent test set. Compared with other published hematoma segmentation methods, the proposed network achieves the best segmentation performance and volume estimation.

4. The volumetric distribution and shape characteristics are extracted from the automated hematoma segmentation. These features are integrated with clinical observations and used to construct a random forest model to predict 6-month mortality. The results show that features extracted from CT scans can greatly improve 6-month mortality prediction. The proposed prediction method yields an average area under the precision-recall curve (AUPRC) of 0.559 and an

average area under the receiver operating characteristic curve (AUC) of 0.853 using 10-fold cross-validation on a dataset comprised of 828 patients. Compared with the widely used IMPACT model [32], the proposed model achieves more than a 5% increase in AUC and 10% in AUPRC.

2.2 Related Work

2.2.1 Related work on hematoma segmentation

Conventional image processing techniques have been applied for hematoma segmentation. In [33], the expectation-maximization was applied on a Gaussian Mixture Model to segment four components, including hematoma regions, normal tissues, white-matter regions, and catheters. In [34], the thresholding technique is used to find the brain tissue and hematoma region clusters. The intensity distribution of pixels is analyzed to segment the hematoma regions. After that, morphological operations were performed as post-processing to get rid of outliers. In [35], a two-class dictionary for normal tissue and hematoma regions was built using patches from "atlas" CT scans and corresponding manual hematoma segmentation. For a given new CT scan, patches were modeled as a combination of the "atlas" patches in the built dictionary to generate hematoma segmentation. The proposed algorithm was evaluated on CT scans from 25 patients with TBI, and the algorithm has a median Dice score of 0.85. A two-stage fully-automated segmentation method has been applied in previous literature for hematoma segmentation on 2D CT slices, including initialization and contour evolving. For the initialization stage, [36] applied a nonlocal regularized spatial fuzzy C-means clustering to segment a coarse hematoma contour. After that, an active contour without edges method was used to refine the contour. CT scans from 30 subjects with different hematoma sizes, shapes, and locations were used to evaluate the proposed method, and a Dice score of 0.92 was produced. Similarly, in [37], the output from fuzzy C-means clustering was used as the initialization of the modified version of DRLSE. In [38], fuzzy c-means clustering and entropy-based thresholding with morphological operations were used to initialize the DRLSE model. The algorithm was developed and tested on CT scans from 35 patients with ICH and achieved an average Dice score of 0.93.

DL techniques have been widely applied in medical image analysis. [39] proposed a semi-supervised multi-task attention-based UNet model. The segmentation task was performed by a UNet architecture and trained using the labeled dataset. The encoder part of the UNet was shared for the unsupervised model, which was trained using the unlabeled dataset to reconstruct the foreground and the background. [40] applied a 3D CNN to segment the chronic type of SDH in pre- and post-operative CT scans. [41] compared the DL segmentation performance with human inter-rater and intra-rater variability. A 3D UNet was applied for spontaneous intracerebral hemorrhage segmentation, and patches from 3D CT scans were used as input to the network. From the experimental results, the proposed algorithm achieved a comparable level to the observer variability. In [31], a mask R-CNN architecture was applied with a custom feature extractor combined with 3D and 2D contextual information. The detected bounding boxes for hematoma regions were further segmented by a segmentation branch.

2.2.2 Related work on outcome prediction based on admission data

In [32], prognostic models were developed using logistic regression models and variables available at admission to predict the mortality and unfavorable outcome in 6 months after the injury. The proposed model was built based on baseline variables, including demographic data, clinical severity scores, CT findings, and biochemical variables. The prognostic models achieved discrimination between patients with good and poor 6-month outcomes after the injury. In this study, the CT characteristics were qualitative and provided by human reviewers. In [42], the role of CT characteristics in outcome prediction for the patient with moderate to severe TBI was investigated by grading the CT scans according to the Rotterdam CT score [43, 44]. In [45], quantitative CT features extracted from automated image analysis were used in outcome prediction. From their experimental results, quantitative CT features were significantly more predictive than qualitative CT features. A major limitation of this study is that the sample size is only 115. CT scans were all from patients admitted to the neurosurgical intensive care unit of the same hospital, which indicates a bias to patients with severe TBI, and the CT scans have low variability.

2.3 Dataset

CT scans from Progesterone for Traumatic Brain Injury, Experimental Clinical Treatment (PROTECT) III trial were used in this study [46]. The PROTECT III trial was conducted at 49 trauma centers in the United States. Study inclusion criteria included adult patients with moderate, moderate-to-severe, or severe TBI, with a GCS score of 4 to 12. Moreover, patients were only enrolled if the study treatment could be initiated within 4 hours after injury. Patients were excluded if the team determined that the patients were non-survivable; had bilateral dilated, unresponsive pupils; or if the patients had physiological findings of hypoxemia, hypotension, spinal cord injury, or status epilepticus [46]. In total, the dataset contains CT scans and clinical assessments of 882 patients. Of the 882 patients enrolled in the clinical trial, the dataset was missing mortality or CT scans from 54 patients. Consequently, 828 patients were included in the mortality classification. The demographic and baseline clinical characteristics of those patients are given in Table 2.1. In total, 676 of 828 patients survived after 6 months. The clinical characteristics of patients in the two groups (survival vs. mortality) are also presented in Table 2.1.

To develop the hematoma segmentation algorithm, 120 CT scans from different

Characteristics	Survival (n=676)	Mortality (n=152)
Age, years		
Median (25th-75th percentile)	32 (23-47)	55 (37.5-68)
Gender, n (%)		
Male	500 (74.0)	106 (36.2)
Race, n (%)		
Caucasian	513 (75.9)	115 (75.7)
African American	105 (15.5)	17 (11.2)
Asian	22 (3.3)	15 (9.9)
American Indian	5 (0.7)	2 (1.3)
Others or unknown	31 (4.6)	3 (2.0)
Cause of Injury		
Motor vehicle accident	272 (40.2)	30 (19.7)
Motorcycle, or scooter accident	132 (19.5)	33 (21.7)
Pedestrian struck by moving vehicle	78 (11.5)	28 (18.4)
Fall	95 (14.1)	37 (24.3)
Assault	36 (5.3)	10 (6.6)
Others or unknown	63 (9.3)	14 (9.2)
Best motor response, n (%)		
None/Extension	54 (8.0)	22 (14.5)
Flexor response	77 (11.4)	28 (18.4)
Withdrawal	224 (33.1)	44 (28.9)
Localizes pain/Obeys commands	321 (47.5)	58 (38.2)
Pupillary reactivity, n (%)		
Bilateral pupil response	91 (13.5)	34 (22.4)
Unilateral pupil response	566 (83.7)	102 (67.1)
No pupil response	19 (2.8)	16 (10.5)
Marshall Score		
I	104 (15.4)	3 (2.0)
II	233 (34.5)	16 (10.5)
III/IV	4 (0.6)	3 (2.0)
V/VI	335 (49.6)	130 (85.5)
Existence of SAH, n (%)		
Yes	427 (63.2)	128 (84.2)
Existence of EDH, n (%)		
Yes	89 (13.2)	32 (21.1)
Glucose (mmol/l)		
Median (25th-75th percentile)	8.0 (6.6-9.4)	8.3 (7.1-9.8)
Hb (g/dl)		
Median (25th-75th percentile)	13.9 (12.7-14.9)	13.1 (11.9-14.5)
iGCS		
Median (25th-75th percentile)	8 (6-10)	7 (6-9)
GOS		
Median (25th-75th percentile)	6 (4-7)	1 (1-1)

Table 2.1: Characteristics of patients.

patients were first randomly selected. The boundary of brain hematoma was manually annotated by a radiologist. The annotated cases were randomly divided into Set 1 (100 cases) and Set 2 (20 cases). A 5-fold cross-validation was performed using Set 1 to evaluate the superiority of the proposed segmentation algorithm. Next, using all of Set 1 as the training set and Set 2 as an independent test set, the proposed hematoma segmentation framework was compared with other published methods.

For mortality prediction, all patients with available CT scans and mortality information in the PROTECT dataset were used. For cases in Set 1, the automated hematoma segmentations were generated using models from the 5-fold cross-validation. For other cases, the automated hematoma segmentations were generated from the model using all of Set 1 as the training set. Features from the hematoma segmentation were extracted and combined with essential clinical observations for mortality prediction. To evaluate the predictive power of the hematoma-relevant features, 10-fold cross-validation was repeated 50 times to avoid bias from the training and test data split.

2.4 Methods

2.4.1 Pre-processing

Let us define a CT raw image of size $H \times W$ as $I_{raw} : \Omega_{img} \to \mathbb{R}$, where $\Omega_{img} = \{1, 2, \ldots, H\} \times \{1, 2, \ldots, W\}$. I_{raw} is an image with gray values stored in Digital Imaging and Communication in Medicine (DICOM) format. In data pre-processing, the gray value stored in DICOM format is converted to Hounsfield units (HU) by the linear transformation

$$I_{HU} = I_{raw} \times slope + intercept, \tag{2.1}$$

where I_{HU} is the transformed image in HU. The parameters *slope* and *intercept* are respectively the rescale slope and rescale intercept retrieved from the DICOM header file. The contrast is then adjusted by choosing a HU range of interest $[a, b]$:

$$I(i,j) = \begin{cases} 0 & \text{if } I_{HU}(i,j) < a \\ \frac{I_{HU}(i,j)-a}{b-a} \times 255 & \text{if } a \leq I_{HU}(i,j) \leq b \\ 255 & \text{if } I_{HU}(i,j) > b \end{cases}, \tag{2.2}$$

where $I(x)$ is the intensity after contrast adjustment at location $x \in \Omega_{img}$. $a = 0$ HU and $b = 80$ HU are commonly used in practice to visualize brain CT images. In this study, as acute hematomas are brighter than normal brain tissues, $a = 0$ HU and $b = 140$ HU were used as a baseline to capture more pathological tissue. CT images adjusted using different values of a and b were used in the mixed loss function, which will be discussed later. After contrast adjustment, the orientation of the 3D brain object was calculated, and volume rotation performed, to ensure the same orientation for all cases [47].

2.4.2 Multi-view CNN architecture

Figure 2.1 depicts the architecture of the proposed Multi-view CNN. We now introduce some mathematical notation that will be utilized throughout this section. The input to the proposed architecture is a 2D adjusted CT image I after pre-processing, and the output is a probability map $O : \Omega_{img} \rightarrow [0, 1]$, where the value in each location gives the probability of the corresponding pixel belonging to a hematoma. The ground truth for the network is a binary annotation mask $L : \Omega_{img} \rightarrow \{0, 1\}$, where one means that the corresponding pixel belongs to a hematoma, and zero that the pixel belongs to a normal region. $F : \Omega_{map} \rightarrow [0, 1]$ denotes a feature map for a convolutional layer, where $\Omega_{map} = \{1, 2, \ldots, H\} \times \{1, 2, \ldots, W\} \times \{1, 2, \ldots, C\}$, with C the number of features used in the convolutional layer.

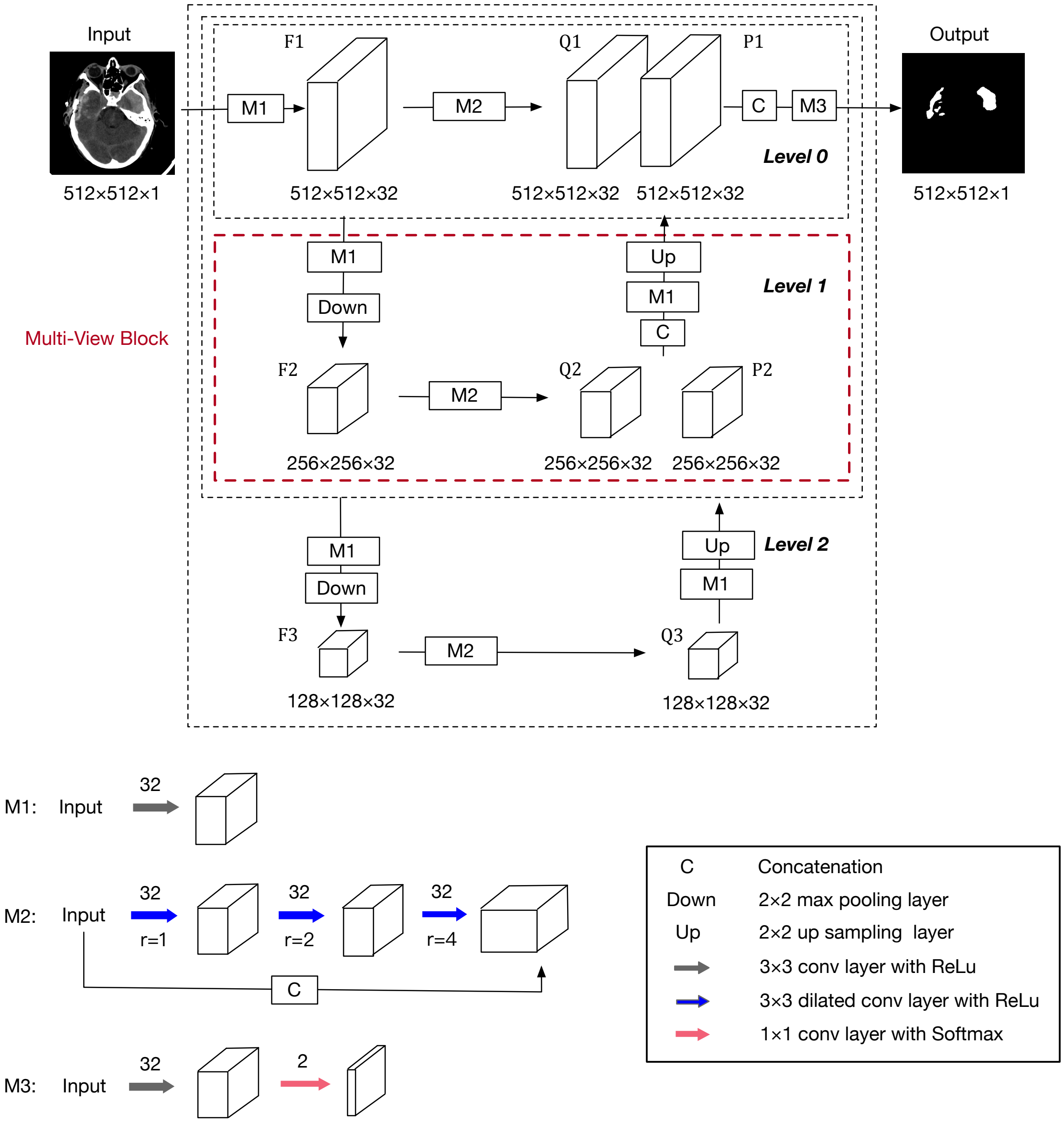


Figure 2.1: Multi-view CNN architecture.

Three modules denoted as M1, M2, and M3 are involved in the network. M1 consists of two convolutional layers with rectified linear units (ReLUs). Let us denote a filter of size $(2d+1) \times (2d+1)$ as $k : \Omega_f \rightarrow \mathbb{R}$, where $\Omega_f = [-s, s]^2 \bigcap \mathbb{Z}^2$, $s \in \mathbb{Z}^+$, $d \in \mathbb{Z}^+$. The convolution operator $*$ between an image I and a filter k can be written as:

$$(I * k)(\mathbf{p}) = \sum_{\mathbf{s}+\mathbf{t}=\mathbf{p}} I(\mathbf{s})k(\mathbf{t}), \tag{2.3}$$

where $\mathbf{t} \in \Omega_f$ and $\mathbf{s}, \mathbf{p} \in \Omega_{img}$. Filters in M1 are all of size 3×3.

Unlike M1, M2 consists of three consecutive dilated convolutional layers [48], with respective dilation rates of 1, 2, and 4. The dilated convolution is defined as

$$(I * k)(\mathbf{p}) = \sum_{\mathbf{s}+r\mathbf{t}=\mathbf{p}} I(\mathbf{s})k(\mathbf{t}), \tag{2.4}$$

where r is the dilation factor, and $r \in \mathbb{Z}^+$. Filters in M2 are also of size 3×3. In M2, a long skip layer as proposed in [49] is used to fuse lower-level features with higher-level ones by concatenating the input of M2 with the output from the last dilated convolutional layer.

M3 is an output module consisting of three regular convolutional layers, with filters in the first two layers having a size of 3×3, and the last one having a 1×1 filter. A pixel-wise softmax activation function follows to generate the probability map.

The backbone of the architecture is shown in Figure 2.1, which provides an overview of multi-view integration. The input image I is first fed into an M1 block, resulting in feature maps F_1. As the scale of extracted features is associated with the size of the convolution operator's receptive field, M1 is used to extract lower-level features, which are shared with subsequent paths. F_1 is an input to both an M2 block and another M1 block. The M2 block can be regarded as a higher-level feature extractor following F_1, where consecutive dilated convolutional layers enlarge the theoretical size of the receptive field of each element in Q_1 to 23×23[1]. The feature maps Q_1 include local features such as the intensity heterogeneity of local image patches, and shape characteristics of smaller hematomas. F_1 is also used as input for a second M1 in Level 1, after which the spatial size is downsampled to 256×256. Next, F_2 is also an input to both an M2 block in Level 1 and an M1 block in Level 2. With the downsampling layer and the dilation in M2, each element in Q_2 has a receptive field of 51×51. As a result, the feature maps Q_2 contain more contextual information and shape characteristics from larger hematomas. Similarly, F_2 is downsampled. With the M2 block following F_3, each element in Q_3 has a receptive field of 107×107, where anatomical features can be extracted. As dense segmentation tasks require an output with the same resolution as the input, F_3 is upsampled to P_2. Q_2 and P_2 are concatenated to fuse features extracted from different views. Similarly, Q_1 and P_1 are concatenated and fed into the M3 block to generate the final probability map.

In summary, a multi-scale representation of the input is generated hierarchically by downsampling layers, with dilated convolutional layers in M2 blocks extracting features at multiple scales. Further, multi-scale feature maps Q_1, Q_2, Q_3 are fused together to segment hematoma of a variety of shapes and sizes. Shown in Figure 2.1, the modules in the red box can be repeated several times to build multi-view architecture with different levels.

In CT scans from patients with acute TBI, hematoma sizes vary remarkably, ranging from tiny spots (<50 mm^3) to large regions (>50 cm^3). The multi-scale representations created in the proposed architecture can be more powerful in feature extraction. Figure 2.2 (a) gives an illustration of the importance in fusing multi-scale feature maps. In the top path, the input is downsampled by an 8×8 max-pooling layer and convolved with a 5×5 Laplacian filter (an edge detection filter), where the receptive field of each element in the resulting feature map is 40×40. In the

[1] A method for calculating the theoretical size of the receptive field was presented in [48].

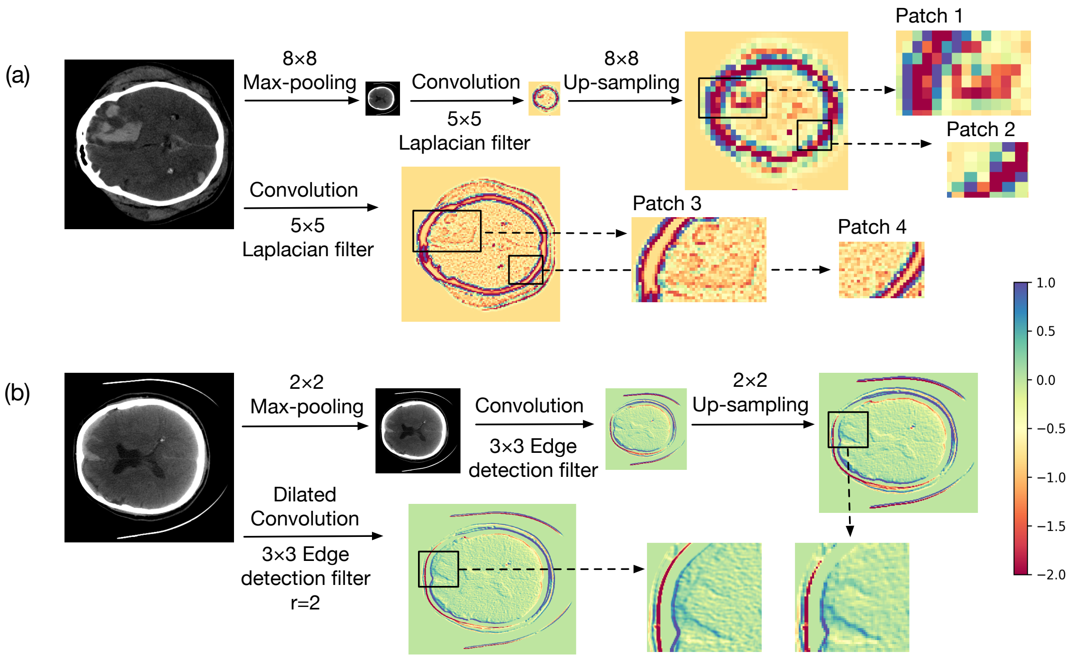


Figure 2.2: Feature map visualization. (a) Comparison of features maps from filters with different receptive fields. (b) Comparison of feature maps from using regular convolution and dilated convolution.

bottom path, the input is directly convolved with the same Laplacian filter, and the receptive field of each element in the corresponding feature map is 5 × 5. The feature map from the smaller receptive field contains fine features and is activated at hematoma boundaries (patch 3) and can detect small hematoma spots (patch 4). In contrast, the feature map from the larger receptive field is much coarser and activated at the location of larger hematomas (patch 1) but may miss smaller hematoma spots (patch 2). In the proposed Multi-view network, feature maps from multiple views can be generated and fused to facilitate both the accurate delineation of hematoma boundaries and localization of hematoma with different sizes.

Compared with directly feeding input images at different scales into the network [50, 51], the proposed architecture uses convolutional layers in M1 to extract lower-level features that can be shared with subsequent layers while avoiding artifacts from downsampling. Additionally, the resulting multi-scale feature maps from the M2 block are gradually fused to build a rich representation of the input image.

The proposed architecture also shares some similarities with U-Net [49], a widely used image segmentation CNN. U-Net uses a number of downsampling layers to enlarge the receptive field and then upsamples the feature maps to recover the resolution. In U-Net, high-level features are extracted with a similar and relatively large receptive field, where smaller object information may be lost. The proposed architecture mitigates this by using a set of M2 modules to generate feature maps at different scales. Additionally, the dilated convolution used in M2 enlarges the receptive fields without sacrificing resolution. Figure 2.2 (b) shows an example of performing a regular convolution and dilated convolution (dilation rate $r = 2$) with an edge detection filter, respectively. Max-pooling and upsampling steps are respectively added before and after the regular convolution to produce the same receptive field as that of the dilated convolution. Comparing the outputs from the two paths, we observe that the dilated convolution produces a finer feature map. In the next section, we will show that the proposed network can achieve both a finer segmentation and higher hematoma identification accuracy as compared to U-Net.

2.4.3 Mixed loss function

In creating an image segmentation mask, a loss function is commonly devised using the intersection and union of the ground truth and the produced segmentation. Given an image I_{HU}, I_0 is generated from image enhancement in (2.2) with $a = 0$ and $b = 140$. With I_0 as an input, the CNN will generate a probability map O_0. Using O_0 and the annotated hematoma mask L, the regular loss for a single image can be written as:

$$loss = -2 * \sum_{x \in \Omega_{img}} \frac{L(x) O_0(x)}{L(x) + O_0(x)} \tag{2.5}$$

In this study, the goal is to build an acute hematoma segmentation system that is robust to CT scans from multiple health centers with different acquisition and imaging protocols, as well as patients under different conditions. Under different imaging settings and patient conditions, the brightness of hematomas as compared to normal regions in CT scans can be slightly different. For example, in the first 0-4 hours, a 2–4% increased absorption of water in the affected brain regions can decrease the hypodensity of tissues with hematoma in the range of $2 - 8$ HU [52]. The subtle differences in contrast between normal and hematoma tissues usually are not visually apparent, and radiologists can still recognize the hematomas with a visual inspection. However, the hematoma segmentation from a trained CNN can be sensitive to contrast differences (as discussed further in §2.5.1). To improve the stability of the CNN, a mixed loss function is proposed by weighting the metrics over images from different contrast enhancements.

First, images $\{I_0, I_1, I_2, \cdots, I_N\}$ are generated with different contrast settings for I_{HU} by introducing random noise into the contrast enhancement. N is an arbitrary number denoting the total number of images of random contrast enhancement. While I_0 is generated with a fixed value $a = 0$ HU and $b = 140$ HU, I_i ($i \in \{1, 2, \cdots, N\}$) is enhanced using $a = 0 + c_i$, and $b = 140 + d_i$, where c_i, d_i are two noise sample from a uniform distribution $\mathcal{U}(-Q, Q)$, where Q is a positive value and controls the magnitude of noise. After a forward pass, hematoma probability maps $\{O_0, O_1, O_2, \cdots, O_N\}$ are generated by the proposed network. To encourage model stability, a mixed loss function was formulated to reduce the error in probability maps resulting from images with different contrasts. The mixed loss function can be written as

$$loss_{mix} = -2 * \sum_{i=0}^{N} w_i \sum_{x \in \Omega_{img}} \frac{L(x)O_i(x)}{L(x) + O_i(x)}, \tag{2.6}$$

where w_i is a weighting factor.

Unlike data augmentation, which assumes that the label of a sample is invariant to transformations, changing the contrast may make hematoma tissues less distinguishable from healthy tissues, while images generated using a, b values significantly different from the original ones may be less plausible. To prevent introducing wrong information into the training process, the contribution of images with different contrast enhancements are adjusted by comparing the similarity between those images and I_0. The similarity between I_i and I_0 is calculated based on the magnitude of c_i and d_i. The weighting factor for I_i and O_i can be written as:

$$w_i = 1 - \frac{|c_i| + |d_i|}{2Q}, \tag{2.7}$$

where $2Q$ is the width of the uniform distribution used to generate random noise. By visually comparing the appearance of the images after enhancement, $Q = 30$ was chosen.

2.4.4 Balanced sampling

In general, there are far fewer CT images with hematoma than those without. The loss function formulations (2.5), (2.6) are only effective when hematoma exists in the input image. To utilize images without hematoma, the training dataset was divided into positive and negative pools based on annotations, where the positive pool contains all images with hematoma and the negative pool contains all those without. In each training step, B images are randomly selected from each pool and fed into the network, where B is the batch size for each pool. The calculated probability maps for those images can be concatenated as a volume $VO : \Omega_v \to [0, 1]$, where $\Omega_v = \{1, 2, \ldots, 2B\} \times \{1, 2, \ldots, H\} \times \{1, 2, \ldots, W\}$. The hematoma masks can also be concatenated and denoted as $VL : \Omega_v \to \{0, 1\}$. The loss function in (2.5) in then modified to

$$loss = -2 * \sum_{x \in \Omega_{img}} \frac{VL(x)VO(x)}{VL(x) + VO(x)}. \tag{2.8}$$

Similarly, the mixed loss in (2.6) is changed to

$$loss_{mix} = -2 * \sum_{i=0}^{N} w_i \sum_{x \in \Omega_{img}} \frac{VL(x)VO_i(x)}{VL(x) + VO_i(x)}. \tag{2.9}$$

2.4.5 Quantitative hematoma feature extraction

Using the probability maps output from the CNN, predicted hematoma masks are generated by assigning each pixel the label with the highest probability. To explore the predictive power of qualitative and quantitative assessment of hematomas, a number of features were extracted and used to build a random forest model for predicting 6-month mortality.

Previous studies have shown that the volume and location of hematomas are important to a patient's outcome. In this study, a 3D volume registration is used to estimate location maps for each patient. The head CT scan from one healthy subject is used as a template, with the annotation of anatomical brain regions including the frontal lobe, temporal lobe, parietal lobe, occipital lobe, and posterior fossa being manually drawn by a radiologist. Examples of the template and annotated location maps are provided in Figure 2.3.

Volume registration was performed using the Elastic toolbox [53] to find the mapping between CT scans from two patients. Let us denote brain CT scans from the healthy subject as $X_{ref} : \Omega_{ref} \to \mathbb{R}$, $\Omega_{ref} = \{1, 2, \ldots, D_{ref}\} \times \{1, 2, \ldots, H_{ref}\} \times \{1, 2, \ldots, W_{ref}\}$, where D_{ref} is the number of slices and H_{ref}, W_{ref} are the height and width of the slices, respectively. The annotation of anatomical brain regions

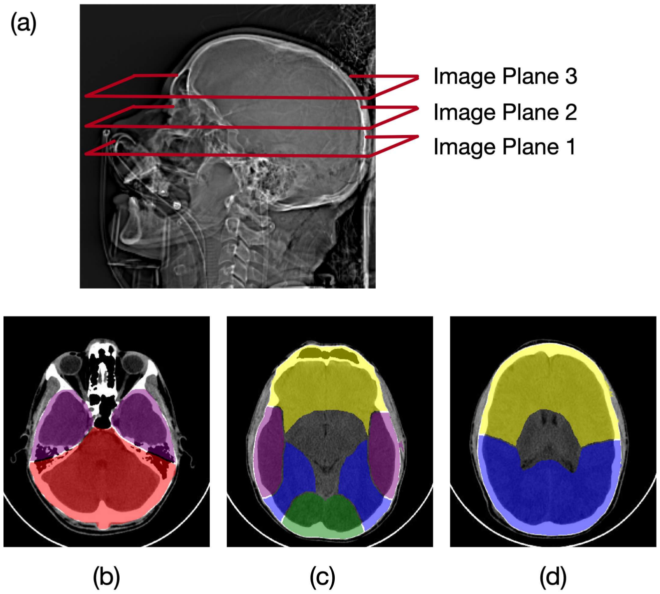


Figure 2.3: Examples of annotated anatomical regions. (a) Lateral view of the skull in CT; (b)-(d): Annotated anatomical regions for image planes 1,2, and 3, respectively. Red: posterior fossa; purple: temporal lobe, yellow: frontal lobe, green: occipital lobe, blue: parietal lobe.

for X_{ref} can be denoted as $Y_{ref} : \Omega_{ref} \rightarrow \{0, 1, 2, 3, 4\}$, where 0–4 correspond to frontal lobe, temporal lobe, parietal lobe, occipital lobe, and posterior fossa, respectively. Given a brain CT scan from another patient $X_s : \Omega_s \rightarrow \mathbb{R}$, $\Omega_s = \{1, 2, \ldots, D_s\} \times \{1, 2, \ldots, H_s\} \times \{1, 2, \ldots, W_s\}$, the coordinate transform from X_s to X_{ref} can be denoted as $T : \Omega_s \rightarrow \Omega_{ref}$, which represents the spatial mapping of every point in X_s to a position in X_{ref}. In this study, an affine transform was used along with the normalized correlation coefficient (NCC) as a similarity measure. The optimal T is estimated by maximizing the similarity between X_{ref} and $T \circ X_{ref}$:

$$\hat{T} = \arg\max_{T} \mathrm{NCC}(X_{ref}, T \circ X_s) \tag{2.10}$$

$$\text{NCC}(X_{ref}, T \circ X_s) = \frac{\sum_{x \in \Omega_{ref}} (X_{ref}(x) - m_1)(T \circ X_s(x) - m_2)}{\sqrt{\sum_{x \in \Omega_{ref}} (X_{ref}(x) - m_1) \sum_{x \in \Omega_{ref}} (T \circ X_s(x) - m_2)}}, \tag{2.11}$$

$$m_1 = \frac{1}{|\Omega_{ref}|} \sum_{x \in \Omega_{ref}} X_{ref}(x) \tag{2.12}$$

$$m_2 = \frac{1}{|\Omega_{ref}|} \sum_{x \in \Omega_{ref}} T \circ X_s(x) \tag{2.13}$$

With the optimized $\hat{T}$, the unknown location map Y_s can be estimated as:

$$\hat{Y_s} = \hat{T}^{-1} \circ Y_{ref} \tag{2.14}$$

With volume registration, location maps can be estimated for every patient in the dataset. Using hematoma segmentations from the CNN and estimated location maps, six volume features are extracted, including hematoma volume in the frontal lobe, temporal lobe, parietal lobe, occipital lobe, and posterior fossa, as well as total head hematoma volume.

Previous studies also found that irregular hematomas such as hematomas with pleomorphic contour, separated adjacent hematomas, and multi-centric hematomas are related to poor outcomes [18]. To incorporate this information into the model, the convexity of the largest hematoma, intensity heterogeneity of the largest hematoma, and the number of hematomas are extracted as shape features from each segmentation.

Category	Feature list
IMPACT core model	Age, Motor score, Pupillary reactivity
IMPACT extended model	Age, Motor score, Pupillary reactivity, Existence of hypoxia†, Existence of hypotension†, Marshall CT classification, Existence of traumatic SAH, Existence of EDH
IMPACT lab model	Age, Motor score, Pupillary reactivity, Existence of hypoxia†, Existence of hypotension†, Marshall CT classification, Existence of traumatic SAH, Existence of EDH, Glucose concentration, Hb concentration
IMPACT without CT features	Age, Motor score, Pupillary reactivity, Existence of hypoxia†, Existence of hypotension†, Glucose concentration, Hb concentration
Volume	Hematoma volume in frontal lobe, temporal lobe, parietal lobe, occipital lobe, posterior fossa; total hematoma volume
Shape	The number of hematomas, convexity of the largest hematoma, intensity heterogeneity of the largest hematoma

Table 2.2: Feature sets for 6-month mortality prediction.

† The feature "existence of hypoxia" and "existence of hypotension" are used in IMPACT models. However, in this study, these two features have no predictive power because patients with hypotension and hypoxemia were excluded in the enrollment stage by PROTECT trial study group [46].

2.4.6 Mortality prediction

A number of clinical observations have been found to correlate with mortality in TBI patients. The three IMPACT [32] models are prognostic models of outcome prediction for patients with TBI that have been widely used and validated. The IMPACT core model includes age, GCS, and pupillary reactivity as core features. The IMPACT extended model includes core features, information on secondary insults and CT findings. The IMPACT laboratory model includes all features in the extended model plus blood hemoglobin and glucose concentrations. These features are patient characteristics that could be determined easily and reliably within the first few hours after injury. Full lists of features used in the three IMPACT models are given in Table 2.2. The laboratory IMPACT model includes manually evaluated qualitative and semi-quantitative measurements from CT scans, such as the Marshall CT classification, presence of traumatic SAH or EDH. To explore the predictive power of quantitative hematoma characteristics from automated hematoma segmentation, an IMPACT feature subset was created by removing CT scan features in the IMPACT full feature set ("IMPACT without CT features" in Table 2.2).

In this study, two types of baselines were built for predicting 6-month mortality. One baseline combines the original IMPACT models with logistic regression, while the other constructs a random forest model on the"IMPACT lab model feature list" shown in Table 2.2. After that, the "IMPACT without CT features" subset was combined with quantitative hematoma features derived from hematoma segmentation (i.e., the volume and shape features in Table 2.2). Random forest models were trained on the combined feature sets, and the prediction performances were compared with the two baselines. To avoid bias from data splitting, 10-fold cross-validation was performed. AUPRC, AUC, F1 score, sensitivity, specificity, and precision were used for model evaluation and comparison. The average value and standard derivation of evaluation metrics were also calculated.

Model		Dice	Jaccard	ICC
Proposed Models with Regular Loss	Multi-view Level 0	0.546 (0.042)	0.365 (0.054)	0.724 (0.033)
	Multi-view Level 1	0.608 (0.015)	0.458 (0.015)	0.925 (0.008)
	Multi-view Level 2	**0.669 (0.019)**	**0.523 (0.027)**	**0.953 (0.018)**
	Multi-view Level 3	**0.669 (0.024)**	0.521 (0.033)	0.952 (0.019)
	Multi-view Level 4	0.665 (0.018)	0.518 (0.024)	0.937 (0.020)
U-Net with Regular Loss	U-Net Level 3	0.642 (0.020)	0.494 (0.022)	0.911 (0.010)
	U-Net Level 4	**0.650 (0.029)**	**0.508 (0.019)**	**0.934 (0.014)**
	U-Net Level 5	0.640 (0.014)	0.500 (0.014)	0.929 (0.017)
Mixed Loss	**Multi-view Level 2**	**0.675 (0.020)**	**0.529 (0.018)**	**0.959 (0.013)**
	U-Net Level 4	0.660 (0.029)	0.517 (0.025)	0.952 (0.013)

Table 2.3: Segmentation performance comparison between Multi-view networks and U-Net from a 5-fold cross-validation on Set 1. The average value and standard deviation from five folds are given. ICC is the intraclass correlation coefficient.

2.4.7 CNN configurations

The CNN was implemented using the TensorFlow library (v.1.10) and trained on an NVidia Tesla V100. Random left-right flipping and the elastic transformation were used to augment our training data. The Adam optimizer was chosen with a learning rate of 10^{-3} to minimize loss. The model was trained for 20,000 steps with a batch size B of 1 for both positive and negative pools. The hyper-parameters of the CNN models were determined using a second 5-fold cross-validation on the first fold of Set 1. Different combinations of the learning rate, batch size, and training steps were tested, and the optimal set of hyper-parameters was chosen using the average Dice coefficient.

2.5 Results and Discussion

2.5.1 Network performance comparison on Set 1

As discussed in §2.4.2, the Multi-view block in Figure 2.1 can be repeated multiple times. Shown in Figure 2.1, the Multi-view Level 0, Multi-view Level 1, and Multi-view Level 2 networks are built with 0, 1, and 2 Multi-view blocks, respectively.

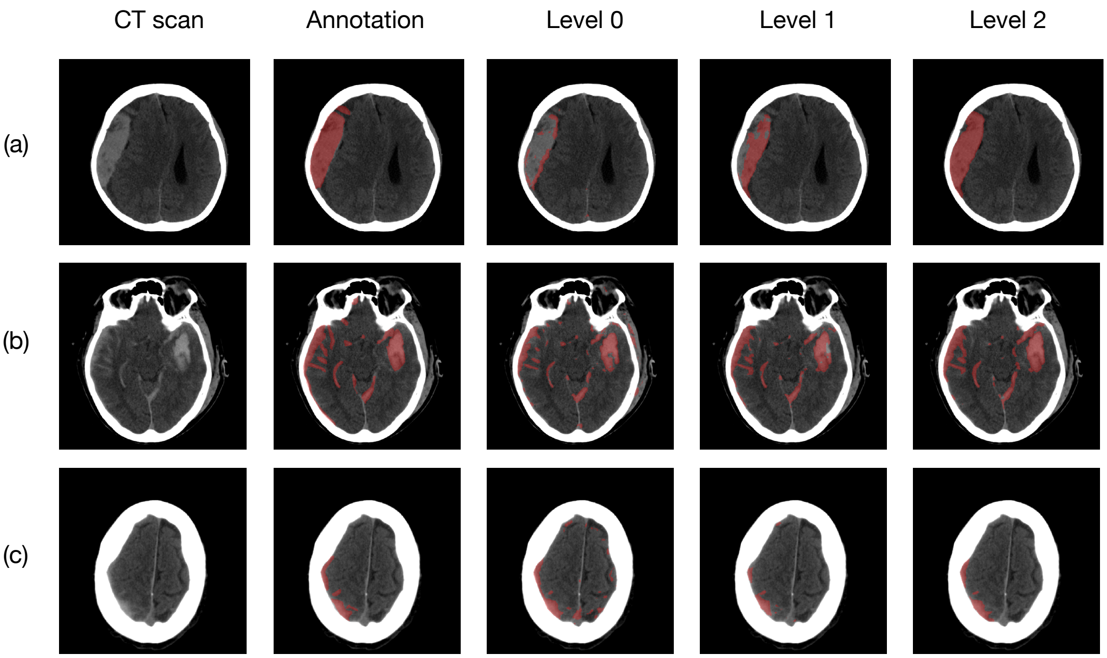


Figure 2.4: Comparison of hematoma segmentation from Multi-view Level 0, Level 1, and Level 2. Annotated and predicted hematomas are shown in red.

The proposed Multi-view network with different numbers of Multi-view blocks was trained with regular loss and tested with 5-fold cross-validation. From Table 2.3, the segmentation performances are significantly improved from Multi-view Level 0 to Multi-view Level 2, which shows the advantage of fusing feature maps from multiple scales. We can also observe that the performances are similar from Multi-view Level 2 to Multi-view Level 4. This may be because the size of the receptive field in Multi-view Level 2 is large enough to capture sufficient contextual information. The receptive field sizes of the last M2 module in Multi-view Level 3 and Multi-view Level 4 are 219 × 219 and 443 × 443, respectively. Considering that the spatial size of the CT scan is usually 512 × 512, features extracted from that broad a receptive field may not add much value to hematoma segmentation.

In Figure 2.4, segmentation results from Multi-view Level 0, Multi-view Level 1, and Multi-view Level 2 are compared. Three example CT images with different types of hematoma are shown. Multi-view Level 0 has very good sensitivity for small hematomas. However, because of the limited size of the receptive field, many false-positive regions exist, and large hematomas cannot be detected very well, which coincides with the analysis in Figure 2.2 (a). With the introduction of the Multi-view block, the hematoma detection specificity increases without missing small hematomas.

We also compared the performance of the proposed architecture with U-Net. The original U-Net contains four downsampling layers (U-Net Level 4). Some variants were also proposed with three or five downsampling layers. From Table 2.3, U-Net Level 4 has the best segmentation performance while the performance of U-Net Level 5 decreases because of the network's depth. In contrast, the additional Multi-view block in Multi-view Level 3 and Multi-view Level 4 does not reduce the segmentation performance as high-level feature maps from multi-scales are extracted hierarchically and then fused with subsequent convolutional layers. From the comparison, the proposed architecture has a better performance than U-Net. Using the multi-view strategy can also help eliminate the careful designing of the depth of the convolutional neural network.

Next, Multi-view Level 2 and U-Net Level 4, which are the best performing networks in their respective categories, were re-trained with mixed loss. From Table 2.3, we can observe that the proposed mixed loss consistently leads to improvement in segmentation performance for both U-Net and proposed Multi-view network. Figure 2.5 presents two examples of segmentation results from Multi-view Level 2 with and without mixed loss. The effect of varying image contrast on hematoma segmentation performance is shown. Given a raw CT scan, three contrast adjustments were performed using $a = 0$ and $b \in \{120, 140, 160\}$. The three resulting images differ slightly in brightness and in the number of artifacts from contrast adjustment. While visual differences across the three contrast adjustments are small, segmentation results from the resultant models trained using regular loss change substantially. In the first example, the segmentation is most accurate when $b = 160$ while the false positive regions

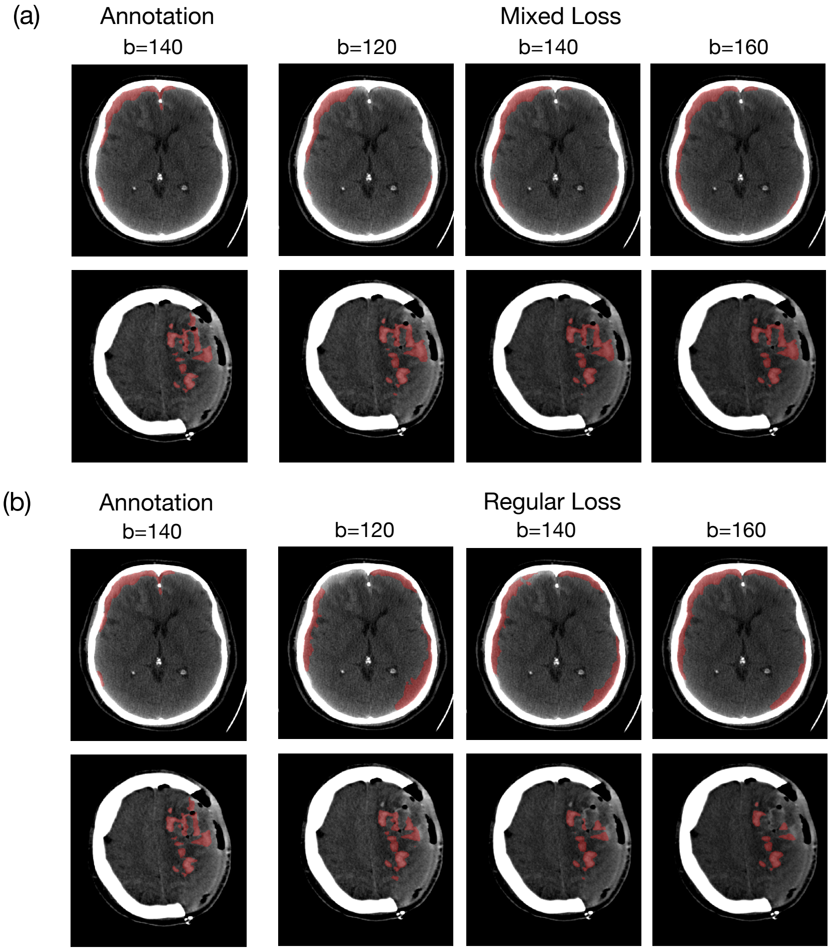


Figure 2.5: Segmentation comparison between results from (a) Multi-view Level 2 with mixed loss and (b) Multi-view Level 2 with regular loss. Images with different contrast were generated with varied b values. Annotated and predicted hematomas are shown in red.

extend significantly when b is decreased. In the second example, the segmentation is most accurate when $b = 120$ while the false negative region appears when b is increased. This may be because the optical window settings to distinguish normal and abnormal tissues are different for these two examples. In contrast, segmentation from the model trained using mixed loss is more consistent when b varies. In the second example, no changes are observed in segmentation with different b values. Using the default value $b = 140$ always leads to accurate segmentation.

Finally, we compared the segmentation performance and volume estimation between the proposed method (Multi-view Level 2 with mixed loss) and U-Net from the

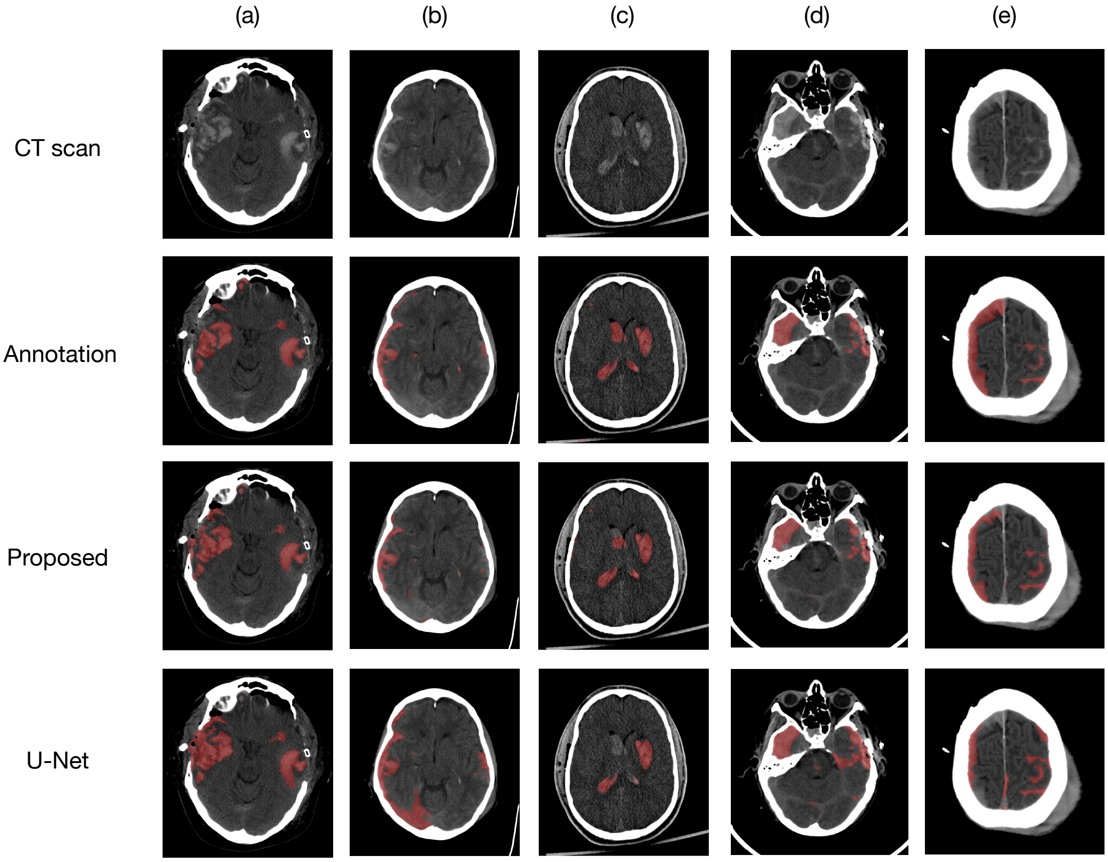


Figure 2.6: Segmentation comparison between the proposed network (Multi-view Level 2 Network with mixed loss) and U-Net (U-Net level 4 with regular loss). Annotated and predicted hematomas are shown in red.

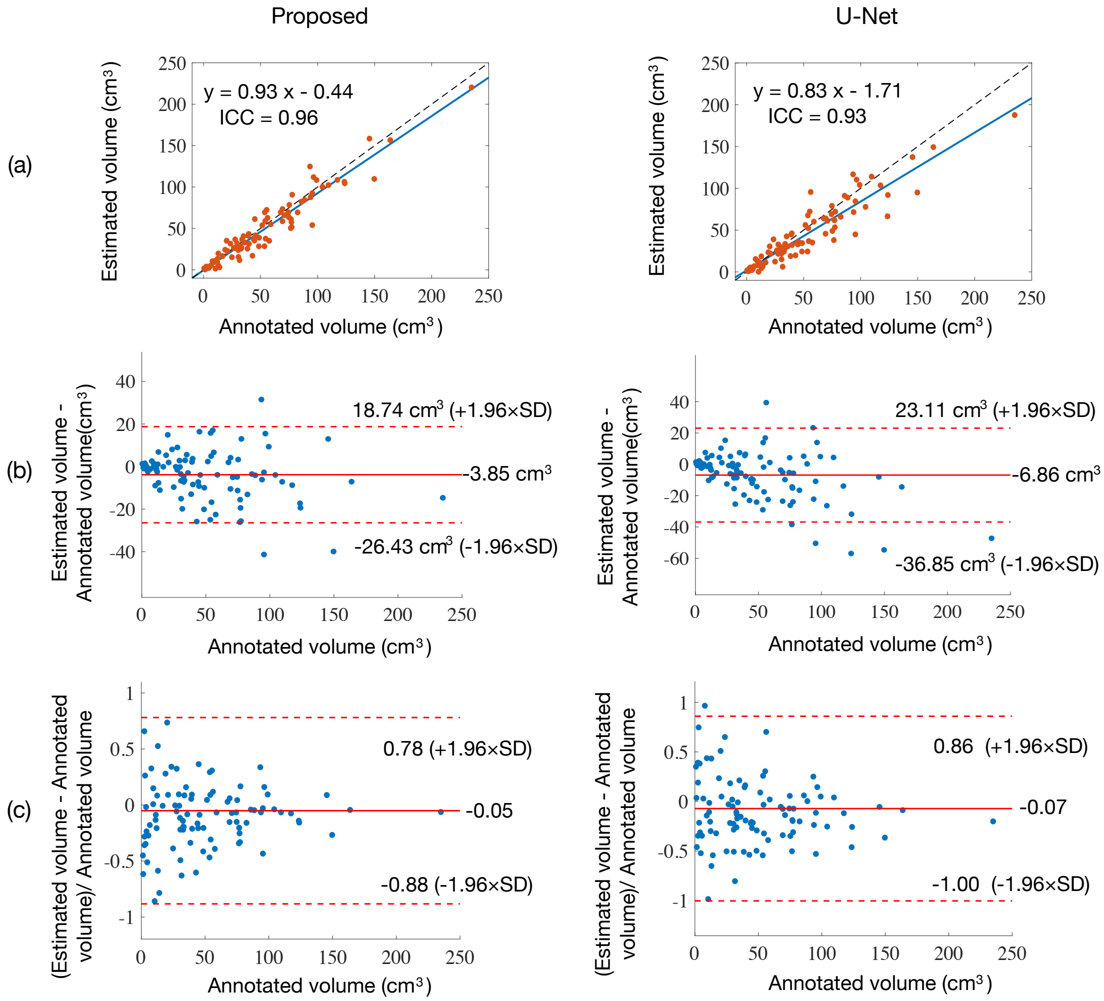


Figure 2.7: Correlation graph and Bland–Altman plot for estimated hematoma volumes using the proposed network (Multi-view Level 2 Network with mixed loss) and U-Net (U-Net level 4 with regular loss). (a) Correlation graph. The blue line is the fitted linear regression line. The black dashed line is the calibration line. ICC is the intraclass correlation coefficient; (b) Bland–Altman plot for volume differences. The mean difference and the standard deviation (SD) of the differences are shown; (c) Bland–Altman plot for the percentage of volume differences of annotated volumes.

previous publication (U-Net Level 4 with regular loss) in Figures 2.6 and 2.7. From Figure 2.6 (a), we can observe that the segmentation from Multi-view Level 2 has a finer boundary while U-Net missed a small hematoma. In Figure 2.6 (b), the brighter region in the occipital lobe is misidentified as a hematoma. In Figure 2.6 (e), the bright region at the midline is misidentified as hematoma, while the real hematoma on the left side is missed. Overall, the proposed network has a higher true positive rate and lower false positive rate for hematoma detection while producing a more refined hematoma boundary delineation.

Figure 2.7 presents the quantitative performance of volume estimation. The segmentation results on the validation sets from the five folds are used to estimate the total hematoma volume for each patient. The hematoma volume is calculated as: $N_p \times \textit{slice spacing} \times \textit{pixel spacing}^2$, where N_p is the number of pixels belonging to hematoma in the patient's CT scan. From Figure 2.7, good agreement between the hematoma volumes from the proposed network's segmentation and those from the annotations can be observed. Segmentation from the proposed network achieves an ICC of 0.96 between the volumes of the predicted hematoma segmentation and volumes of the annotated hematoma segmentation, which is better than the segmentation from U-Net. For the proposed network, the median volume difference is -2.38 (-9.0, 2.0) cm^3, and the median absolute volume difference is 6.12 (2.11, 12.98) cm^3. The bracketed values are the 25^{th} and 75^{th} percentiles, respectively. For U-Net, the median volume difference is -3.72 (-13.94, 1.68) cm^3, and the median absolute volume difference is 6.82 (2.37, 15.23) cm^3. From the Bland–Altman plot, we can observe that the estimated volumes based on the segmentations from the proposed model have lower standard deviation.

Model Description	Dice	Jaccard	ICC
Multi-view+Mixed loss	**0.697 (0.100)**	**0.545 (0.116)**	**0.966**
Multi-view+Regular loss	0.686 (0.112)	0.534 (0.126)	0.964
U-Net [49]	0.654 (0.135)	0.511 (0.142)	0.948
ICHNet [54]	0.635 (0.104)	0.473 (0.107)	0.917
3D U-Net [30]	0.581 (0.155)	0.429 (0.128)	0.891
3D Active Contour [29]	0.496 (0.154)	0.343 (0.132)	0.823

Table 2.4: Hematoma segmentation performance comparison between the proposed method and published methods on Set 2. The average value of evaluation measurements and standard deviation among CT scan cases are given. ICC: intraclass correlation coefficient between the estimated volumes and annotated volumes.

2.5.2 Network performance comparison on Set 2

The Multi-view Level 2 with mixed loss was compared with other published segmentation methods. The results are shown in Table 2.4, where Dice coefficient, Jaccard index, sensitivity, specificity, precision and ICC between the estimated hematoma volumes and annotated hematoma volumes are presented for comparison. All models were trained using all cases in Set 1 and tested on Set 2. From the table, Multi-view Level 2 with mixed loss achieved the best segmentation performance, having the highest Dice coefficient, Jaccard index, and ICC. From our results, the ICHNet [54] and 3D U-Net [30] have a higher sensitivity while precision is significantly lower than other methods. From the results using 3D U-Net, the introduction of 3D contextual information did not improve the overall segmentation performance. This may be due to the dataset being comprised of CT scans collected from multiple centers with varying slice spacing. Training a 3D network on such cases may impair model performance.

2.5.3 Mortality prediction

The results of the 6-month mortality prediction are shown in Table 2.5. One observation from the table is that the AUPRC and F1 are much smaller than AUC.

Model	AUPRC	AUC	F1	Recall	Precision
IMPACT lab model feature list + Volumes	**0.563** **(0.076)**	0.852 (0.038)	0.617 (0.057)	0.684 (0.100)	**0.586** **(0.102)**
IMPACT w/o CT features + Volumes	0.559 (0.072)	**0.853** **(0.034)**	**0.621** **(0.060)**	0.700 (0.107)	0.580 (0.101)
IMPACT lab model feature list	0.498 (0.072)	0.819 (0.038)	0.561 (0.069)	0.677 (0.128)	0.504 (0.108)
IMPACT without CT features	0.441 (0.068)	0.776 (0.046)	0.539 (0.049)	0.639 (0.089)	0.480 (0.080)
original IMPACT core model	0.379 (0.065)	0.798 (0.038)	0.478 (0.045)	0.704 (0.127)	0.381 (0.083)

Table 2.5: Comparison of 6-month mortality prediction using different combinations of feature sets. The last three models (original IMPACT lab model, original IMPACT extended model, original IMPACT core model) were built by logistic regression while the others were built using random forest. The average value of evaluation measurements and standard deviation of the 10-fold cross-validation are given.

This is due to the class imbalance in the dataset, where the ratio of the number of positive samples (mortality) to negative samples (survival) is around 1:4.5.

Among the three original IMPACT models [32], the IMPACT lab model achieved the highest performance, with an average AUPRC of 0.427 and AUC of 0.798. Using “IMPACT lab model feature list”, the random forest model achieved an average AUPRC of 0.498 and AUC of 0.819. As IMPACT features include qualitative and semi-quantitative findings from CT scans, a random forest model was trained using only “IMPACT without CT features”. The classification performance decreased with an average AUPRC of 0.441 and AUC of 0.776, which indicates the predictive power of CT findings.

Next, the predictive power of volume features and shape features derived from the automated hematoma segmentation was explored. From the experimental results, adding either volume features or shape features to “IMPACT without CT features” resulted in significant and consistent improvements in classification performance. Adding volume features to “IMPACT without CT features” resulted in an average AUPRC of 0.559 and AUC of 0.853, while adding shape features to “IMPACT without CT features” resulted in an average AUPRC of 0.539 and AUC of 0.841.

However, when both volumes and shape features were added, no further improvement in classification performance was observed. This may be because hematoma shape is related to hematoma size and growth. A previous study has shown that large hematomas are significantly more irregular in shape and heterogeneous in density [55].

Finally, a model combining "IMPACT lab model feature list" and volume features was trained. The overall classification performance is very close to using the combination of "IMPACT without CT features" and volume features, which indicates that our automated quantitative measurements without manual evaluation can provide sufficient information regarding hematoma. Considering that a ML algorithm seeks to utilize a minimal set of features with sufficient information to perform a classification task, the combination of "IMPACT without CT features" and volume features extracted from the automated hematoma segmentation should be considered the best feature set to construct a 6-month mortality prediction model.

2.5.4 Feature analysis in mortality prediction

To investigate the contribution of features in mortality prediction, we first analyzed the correlation between individual features in "IMPACT lab model features list + Volume features" and mortality. In Table A.1, for features included in "IMPACT lab model feature list", we can observe that patients who do not survive are significantly older, with significantly worse motor response, worse pupillary reactivity, higher Marshall score, more traumatic SAH, and lower Hb concentration. The differences in the existence of EDH and glucose concentration are not significant (if the threshold $p<0.001$ is deemed significant). In Table A.1, patients who do not survive have significantly higher hematoma volumes in each anatomical region.

In addition to the correlation between a single feature and the mortality, we calculated feature importance in the random forest models. Feature importance was

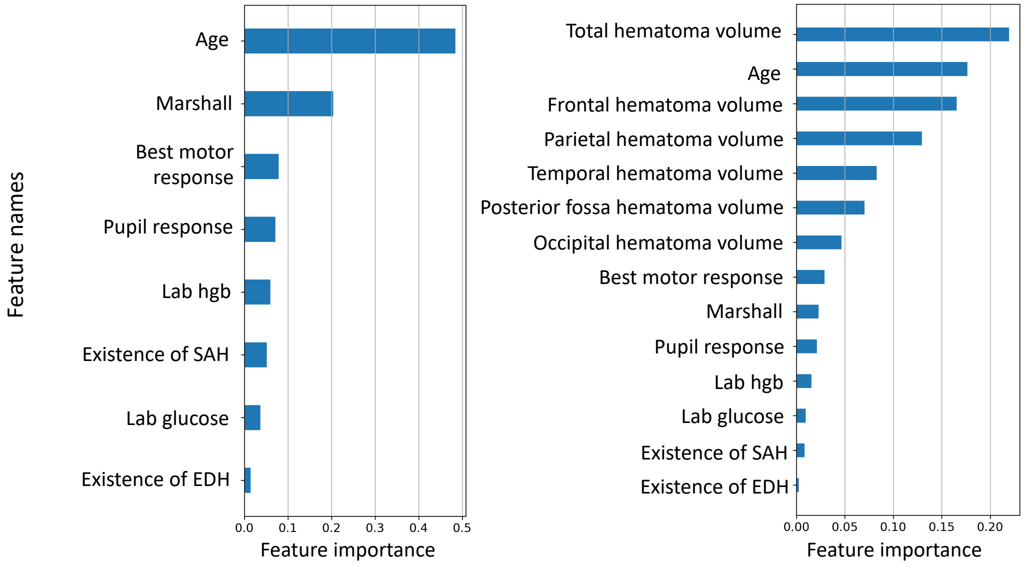


Figure 2.8: Feature importance in the random forest models. (a) Feature importance in the random forest model with “IMPACT lab model feature list”. (b) Feature importance in the random forest model with “IMPACT lab model features list + Volume features”. The importance of “existence of hypotension” and “existence of hypoxia” from “IMPACT lab model feature list” are not shown because patients presenting with hypotension and hypoxemia were excluded from the PROTECT III trial [46].

calculated by how much each feature contributes to decreasing the impurity. Figure 2.8 compares feature importance in the random forest model with “IMPACT lab model features list + Volume features” with that of the random forest model with “IMPACT lab model feature list”. From Figure 2.8, in the random forest model using the “IMPACT lab model feature list”, CT findings such as Marshall CT classification and existence of traumatic SAH have a quite high importance. However, in the random forest model with “IMPACT lab model features list + Volume features”, the importance of all CT findings in ”IMPACT lab model feature list” decreased while Volume features have a higher importance. The feature importance comparison further suggests that our automated quantitative measurements without manual evaluation can provide sufficient and richer information regarding hematoma. From Figure 2.8, we can observe that the epidural hematoma and glucose concentration has very low feature importance in both models, which is consistent with the result in Table A.1 that the differences between the existence of epidural hematoma and glucose concentration are not significant.

The features “existing of hypotension” and “existence of hypoxia” from “IMPACT lab model feature list” were not included in the above feature analysis because patients presenting with hypotension and hypoxemia were excluded from the PROTECT III trial [46].

2.5.5 Limitations

There are some limitations to this study that should be considered when interpreting the results. First, while it is known that specific hematoma types may differentially impact mortality, the current methods are not able to distinguish various hematoma subtypes. Instead, the proposed CNN model segments the total acute blood volume in the brain. We have performed some prior work on developing algorithms that can distinguish specific hematoma subtypes, such as subdural hematoma [56]. In the future, the development of comprehensive methods to separately quantify individual hematoma subtypes may further enhance the predictive accuracy of these methods. Second, although our dataset has the advantage of including patients with moderate and severe TBI from multiple centers using many different types of CT scanners, and that we used 10-fold cross-validation to test the mortality classification models, validation of our findings on additional external datasets remains a necessary next step. Because our dataset is limited to patients presenting to academic medical centers who met eligibility criteria for enrollment in a randomized controlled trial, the results may not generalize as well to the overall population of patients with moderate and severe TBI. Patients with physiological findings of hypotension and hypoxemia were excluded from the PROTECT III trial, so the contribution of these important clinical variables could not be assessed.

CHAPTER III

Automated Feature Extraction from Colonoscopy Videos and Outcome Prediction for Patients with Ulcerative Colitis

3.1 Introduction

Optical colonoscopy is a medical procedure to inspect the mucosal surface to detect abnormalities in the colon. It is an indispensable tool for evaluating many gastrointestinal diseases. During the procedure, a flexible probe with a charge-coupled device camera and a fiber optic light source at the tip is inserted into the rectum and advanced through the colon. After reaching the proximal portion of the intestine (typically the cecum or ileum), the physician withdraws the colonoscope and visually inspects the tissue surface for abnormalities.

Colonoscopy is the primary tool used to screen for colorectal cancer and precancerous lesions and is recommended for all adults over age 50, with repeated exams every 3-10 years based on risk stratification [57, 58]. Additionally, colonoscopy is commonly used to locate and treat sources of lower gastrointestinal bleeding [59, 60]. Colonoscopy is also used to investigate causes of diarrhea, in particular, inflammatory bowel diseases (IBD) [61]. Inspection of the mucosal surface by colonoscopy is particularly important in IBD, where disease severity assessment and monitoring of treatment effectiveness are heavily dependent on the findings of repeated colonoscopies [62, 63, 64]. In this chapter, we focus on patients with UC. However, the proposed colonoscopy video analysis system is expected to be transferable to other colon diseases. Endoscopic evaluation is a principal component of definitions for disease severity and therapeutic response used in both the assessment of investigational medications and the day-to-day decision-making for the patient with UC. Although existing biomarkers, such as fecal calprotectin and histopathologic scoring, provide additional measures of biological disease activity, endoscopy continues to serve as the reference for objective disease assessment. As a result, routine endoscopy to assess disease status is recommended in the recently published American College of Gastroenterology clinical management guidelines, the STRIDE (Selecting Therapeutic Targets in Inflammatory Bowel Disease) international consensus statement, and by regulators in the setting of clinical trials [62, 65, 66].

Mayo endoscopic subscore (MES) is the most commonly used severity score to summarize the entire colonoscopy video because of its simplicity and physician familiarity [67]. The MES is a 4-level scale of severity [range, 0-3] with higher scores reflecting increasing disease severity based on features including erythema, erosions, ulcerations, and bleeding [68]. Beyond assessing therapeutic effect, low or reduced MES scores are associated with a lower risk of future colectomy and clinical relapse [69, 70]. However, conventionally-used endoscopic severity scores such as MES are qualitative and sparse. In addition, human interpretation of colonoscopy videos is time-consuming and subject to inter-observer variation, threatening the accuracy and reproducibility of these important assessments and limiting tracking of the evolution of findings from colonoscopy over time. When asked to grade overall disease severity using endoscopic videos, 10 IBD specialists had 78% agreement when the severe disease was present, but only 37% and 27% agreement for moderate disease and normal, respectively [71].

Thus, there is a need to automate the analysis of colonoscopies to standardize reporting so that disease assessments are uniform regardless of colonoscope operator experience. An automated colonoscopy analysis system holds the potential to extract a more comprehensive patient profile which can be more informative in diagnosis, treatment recommendation, or outcome prediction.

This chapter aims to develop a novel colonoscopy-based CDS system that can extract comprehensive feature representation from colonoscopy videos to evaluate the patient's condition and outcome. An overview of the proposed system is shown in Figure 3.1. To analyze colonoscopy videos, non-informative frame detection, biopsy forceps detection, disease severity score classification, and location estimation for individual frames are proposed. For the disease severity score classification, we adapt the definitions of score levels in MES to grade the severity of individual frames. To avoid confusion, in this chapter, "MES" refers to the conventional MES used for the entire colonoscopy video, and "severity score" refers to the score for the individual frame. Figure 3.2 shows examples of colonoscopy frames with severity score 0 to 3. After the contextual information extraction, a disease severity distribution can be derived over the entire colon. A novel feature representation is extracted from the distribution. Later, decision-making models are built that can estimate MES and the patient's outcome. Our hypothesis is that the agreement between estimated MES and manually annotated MES is close to an inter-observer agreement. And the feature representation from the severity distribution is expected to have a higher predictive value than conventional MES in estimating the patient's outcome.

As far as our knowledge, there is no existing work on the proposed colonoscopy-based CDS system. Image classification has been the focus of existing efforts in automated colonoscopy video analysis, such as informativeness classification and polyps detection. Camera localization is an important component for interpreting findings and calculating severity distribution. Localizing lesions can help generate contex-

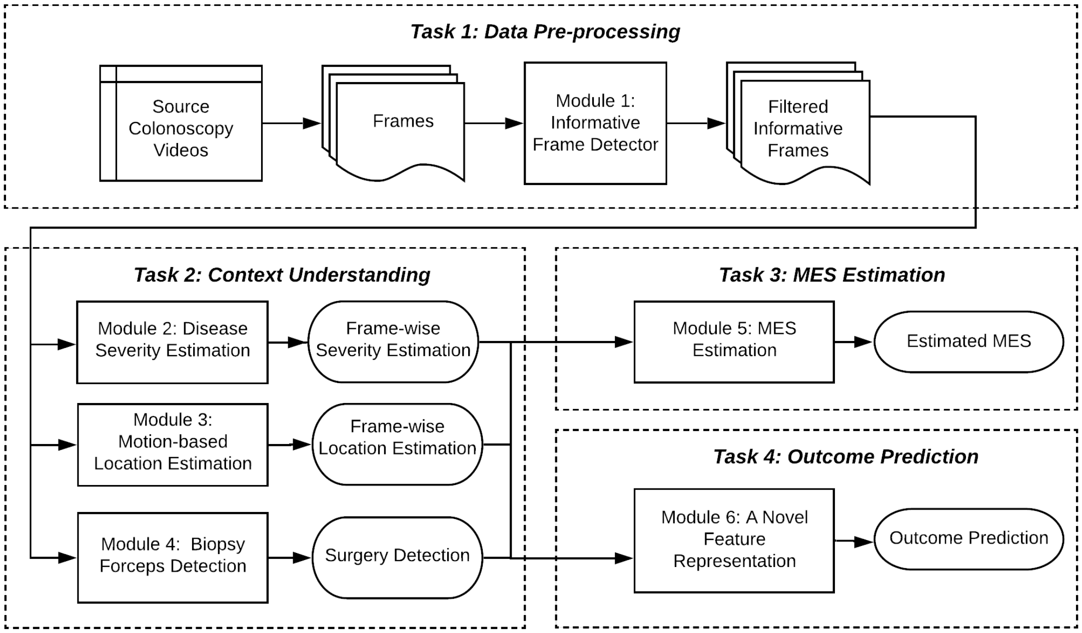

Task 1: Data Pre-processing
Source Colonoscopy Videos
Frames
Module 1: Informative Frame Detector
Filtered Informative Frames
Task 2: Context Understanding
Module 2: Disease Severity Estimation
Frame-wise Severity Estimation
Module 3: Motion-based Location Estimation
Frame-wise Location Estimation
Module 4: Biopsy Forceps Detection
Surgery Detection
Task 3: MES Estimation
Module 5: MES Estimation
Estimated MES
Task 4: Outcome Prediction
Module 6: A Novel Feature Representation
Outcome Prediction

Figure 3.1: An overview of the proposed colonoscopy-based CDS system for patients with UC.

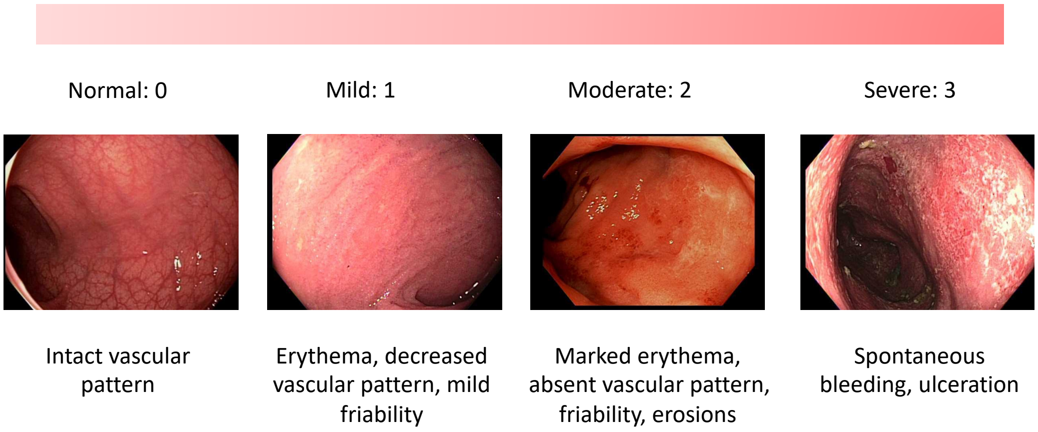

Normal: 0
Mild: 1
Moderate: 2
Severe: 3
Intact vascular pattern
Erythema, decreased vascular pattern, mild friability
Marked erythema, absent vascular pattern, friability, erosions
Spontaneous bleeding, ulceration

Figure 3.2: Examples of frames with graded severity sore 0 to 3 using MES concepts and definitions.

tual information by providing anatomical awareness. The localization of the camera can help improve the accuracy in diagnosis and prognosis of multiple diseases identified by colonoscopy, including cancer position and burden, bleeding source, and IBD distribution [72, 73, 74, 75]. Existing methods for endoscopic object localization fall into two categories: sensor-based localization methods [76] and computer vision-based camera motion estimation methods [77, 78]. While sensor-based methods can provide the absolute position, a relative position of the camera with respect to the end and start of the colon may be sufficient for a contextual understanding of colon features. Sensor-free methods of camera location estimation using computer-vision methods are likely to sacrifice marginally relevant accuracy in exchange for improved feasibility. Sensor-free camera localization methods are attractive as no additional equipment is needed, allowing for rapid integration into clinical workflows, wide availability, and low financial cost.

The main contributions of the work in this chapter are as follows:

1. A novel camera localization algorithm is designed that overcomes the challenges of motion tracking in colonoscopy videos. Significant modifications are made over previous methods (will be discussed in §4.2) to improve the performance of pose estimation in endoscopic videos. With the camera pose estimation, a novel relative location index estimation and anatomical colon segment classification algorithm is devised to provide location awareness for individual frames from colonoscopy videos.

2. A novel contextual understanding method is proposed based on the results from image classification models (non-informative image classification, image severity classification, and biopsy forceps detection) and camera localization algorithm. It is the first time that a spatial disease severity distribution over the entire colon is derived for individual colonoscopy videos. A novel feature representation is extracted from the estimated disease severity distribution to capture the disease

characteristics from a colonoscopy video, with which an MES estimation model and outcome prediction model are built.

3. The proposed camera localization algorithm has been applied to colonoscopy videos collected from routine practice. The videos and frames were annotated by experienced gastroenterologists. The performance of the anatomical colon segment classification is compared with other methods. An additional metric is calculated using data from ScopeGuide® (Olympus Corporation, https://www.olympus-global.com/) to evaluate the relative location estimation, which can provide approximate length information of the inserted scope into the colon. The location index derived from ScopeGuide length is compared with the location index estimated from the proposed localization system. To the best of the authors' knowledge, it is the first time that a localization system has been proposed and evaluated on colonoscopy videos from routine practice.

4. The proposed CDS models has been validated on colonoscopy videos collected from routine practice and clinical trials. Our experimental results support the potential for artificial intelligence to provide endoscopic disease grading in UC that approximates the scoring of experienced reviewers. Based on the feature representation extracted from the disease severity distribution, the outcome prediction model shows a significantly higher performance compared with the one with MES only. It indicates that a more comprehensive patient profile can be extracted to better capture the disease characteristics with the proposed automated colonoscopy video analysis method.

3.2 Related Work

3.2.1 Related work on image classification

Several methods have been proposed for non-informative colonoscopy image classification. Texture analysis using Local Binary Patterns on the frequency domain was presented in [79] to detect the non-informative frames. A set of convolutional neural network (CNN) architectures was explored in [80] and the effectiveness of CNNs in image classification was demonstrated. In [81], non-informative frames were classified through motion, edge, and color features. Most of the previous methods have focused on either hand-crafted feature extraction or end-to-end DL techniques.

Our preliminary work on informative frame classification was described in [82]. In that work, hand-crafted features were combined with bottleneck features for image classification. From the experimental results, the combination of bottleneck features in the RGB color space, and hand-crafted features in the hue-saturation-value color space can boost the classification performance when the size of the training set is small. Later, we found that the DL method alone achieved comparable performance when applying this method to the larger dataset of the current study. As a result, in this study, a CNN was used directly for image classification.

3.2.2 Related work on camera pose estimation

3.2.2.1 Self-supervised learning for camera pose estimation on monocular videos

DL architectures have achieved success in relative camera pose estimation and single view depth estimation [83, 84, 85] on monocular videos. While traditional camera pose estimation algorithms, such as visual odometry, are effective in certain settings, their reliance on accurate image correspondence matching causes problems when the images are of low texture and from a complex environment. DL methods may overcome these challenges by additional supervision.

Based on the view synthesis technique [86], then photometric error between the synthesized new view and real new view can be used as supervision to train the network, which eliminates the requirement for ground truth data. In [87], an end-to-end framework named “SfMLearner” was proposed to jointly train the single view depth and camera pose using projection error. This framework utilized unlabeled data but had a performance comparable with approaches that require ground truth. Authors in [88] proposed a simple normalization of the estimated depth map, which can effectively avoid depth prediction saturating to zero. In addition, a Direct Visual Odometry [89] pose predictor was incorporated into the end-to-end training to establish a direct relationship between the depth map and the camera pose prediction. Joint pose and depth estimations prevent the determination of absolute scale. To recover the absolute scale, UnDeepVO, proposed in [90], was trained using stereo image pairs. After training, the model was tested on consecutive monocular images, yielding good performance on pose estimation.

In the self-supervised framework using photometric loss, several assumptions are implicitly made, including (1) a static scene; (2) Lambertian reflectance, i.e., the brightness is constant regardless of the observer’s viewing angle; (3) no change in lighting between two consecutive frames; and (4) no occlusion between two consecutive frames. These assumptions may fail in real-world applications. A number of studies proposed additional loss terms to improve the robustness of the network in these circumstances. In [91], in addition to photometric loss, deep feature-based warping loss was proposed to take contextual information into consideration rather than per-pixel color matching alone. Salient feature correspondences were extracted in [92], and a matching loss constrained by epipolar geometry was proposed to improve network optimization. GeoNet was proposed in [93] to jointly estimate monocular depth, camera pose, and optical flow. A geometric consistency measurement was proposed as an additional loss term to improve the network's resilience to outliers. The GeoNet model achieved state-of-art performance on the KITTI dataset [94] for all three tasks. In [95], the self-supervised framework was extended by applying the Charbonnier penalty to combine spatial and temporal reconstruction losses. In [96], a geometry consistency constraint was proposed to enforce the scale-consistency of depth and pose networks. Their results showed that the proposed pose network achieves performance commensurate with methods using stereo videos. In [97], a re-estimation approach was proposed where the camera pose estimation was decomposed into a sequence of smaller pose estimation problems. For smaller pose estimation problems, the assumptions made in camera pose estimation algorithms are more likely to hold.

3.2.2.2 Vision-based camera pose estimation in monocular endoscopic videos

Camera motion tracking in endoscopic videos has been investigated in a few studies previously. An optical flow approach to tracking colonoscopy video was proposed in [78]. The focus of expansion (FOE) was calculated using a combination of sparse and dense optical flow calculations. Based on calculated FOE, the computation of the camera's rotation and translation parameters from the optical flow field were separated. This approach is sensitive to optical flow and FOE calculations. In [77], Kanade-Lucas-Tomasi features were extracted and tracked through consecutive frames. After that, a visual odometry algorithm was used to calculate camera motion by assuming each pair of consecutive frames to be a stereo pair. In this approach, the translation estimation is subject to arbitrary scaling and the camera's speed is required.

DL has also been applied to camera pose estimation for endoscopic videos. A CNN was proposed in [98] to estimate the pose of the colonoscope. The network was trained and evaluated on simulated videos. Their results showed that the pose estimation from CNN was more accurate and faster than feature-based computer vision methods. In [99], the depth map was first estimated from images using the Tsai–Shah Shape from Shading method [100]. After that, a recurrent convolutional neural network was applied to model dynamics across the frames and estimate the camera pose. While the evaluation on a real pig stomach dataset showed that the method achieved high translational and rotational accuracy, ground truth for camera pose was also required to train the network. A self-supervised framework has also been applied to endoscopic videos. In [101], SfMLeaner was applied to estimate the motion of the endoscopic capsule robot. Evaluations on videos collected from ex-vivo porcine stomach were used to demonstrate the effectiveness of the method. In [102], a calibration-free framework was proposed by estimating the camera intrinsic parameters. In [103], temporal information among consecutive frames was explored by a long-short-term-memory layer to improve the accuracy of pose estimation. In [104], a self-supervised network was applied to generate pseudo-RGBD frames for endoscopic videos. The camera's ego-motion was estimated using a keyframe-based photometric method and then used for scene reconstruction. In [105], a public dataset and a self-supervised network were developed for camera motion and depth estimation in endoscopic videos, where a spatial attention module was developed to encourage the network to focus on highly textured tissue regions.

3.2.2.3 Motivation of the proposed camera pose estimation algorithm in this chapter

Recent progress in self-supervised camera pose estimation has focused on extending the framework and loss function to improve the network's robustness to violations of the assumptions made by photometric loss. The majority of the methods were trained and validated on the KITTI dataset, which has fewer rotational variations. The self-supervised framework proposed for driving videos has already been successfully applied to endoscopic videos. However, one important limitation of existing work is that they were only validated on either simulated endoscopic videos or a sequence of selected frames from real videos.

The aim of this study is to track the camera within colonoscopy videos. Camera motion tracking in the colon environment can be very challenging because of its complex geometry and a low and similar textural pattern across the colon. From a previous study, a DL-based motion estimation method is more suitable than a feature matching based system [99]. For colonoscopy videos, the motion between two consecutive frames is usually quite small. As such, one can assume the scene is static and lighting changes minimal in the absence of biopsies. However, in the colon environment, the assumption of a Lambertian surface may not hold due to the presence of surface moisture. Specular regions may exist where the brightness can change significantly under different viewing angles. The existence of specular regions impairs the calculation of photometric loss and that of other previously proposed loss terms such as image similarity loss [93] and feature matching-based loss[91].

To overcome this challenge, a specular mask is estimated by extending the network to correct the photometric loss. Additionally, optical flow was added as input and a calculated motion consistency term was used to to improve the robustness and generalizability of the network. As the intrinsic complexity of the colon environment poses several challenges to camera motion estimation, it is necessary to validate the proposed method using real colonoscopy videos from routine practice.

3.3 Datasets

3.3.1 Dataset for non-informative frame classification and biopsy detection

Colonoscopy videos were collected from patients undergoing routine colonoscopy. Each colonoscopy collected was from a unique subject. Videos were recorded at 1920×1080 resolution 10-bit color depth, and 60 frames per second (FPS). All colonoscopies were performed using a CF-HQ190 or PCF-H190 colonoscope and CLV-190 image processors (Olympus Corporation, Inc). Videos are from patients with UC diagnosis that was defined using the following factors: two administrative diagnosis codes for UC (ICD-9 or ICD-10) on two separate encounters, prior histologic UC diagnosis, and the use of at least one UC medication [106].

For non-informative frame classification and biopsy detection, frames were sampled at 1 FPS from 29 colonoscopy videos. In total, the image classification dataset contains 34,810 frames, which were manually annotated by a gastroenterologist. In this study, frames captured in close proximity to the colon wall, with significant motion blur, with over- or under-exposure, or those captured outside of the body were annotated as non-informative. Figures 3.3 (a)-(b) present examples of informative frames while Figures 3.3 (c)-(g) present examples of non-informative frames. 24,425 of 34,810 frames were annotated as non-informative (labeled as "1"), with the median percentage of non-informative frames in each colonoscopy video being 59.8%. For biopsy forceps detection, 932 frames were manually annotated as "with biopsy forceps". The median percentage of frames with biopsy forceps is 3.0%. Examples of frames with biopsy forceps are shown in Figures 3.3 (h)-(i). The 29 colonoscopy videos in this dataset were randomly split into the training set ($n = 19$) and test set ($n = 10$). The training set was used for hyper-parameter tuning and model training. The test set was used to evaluate the performance of the trained models.

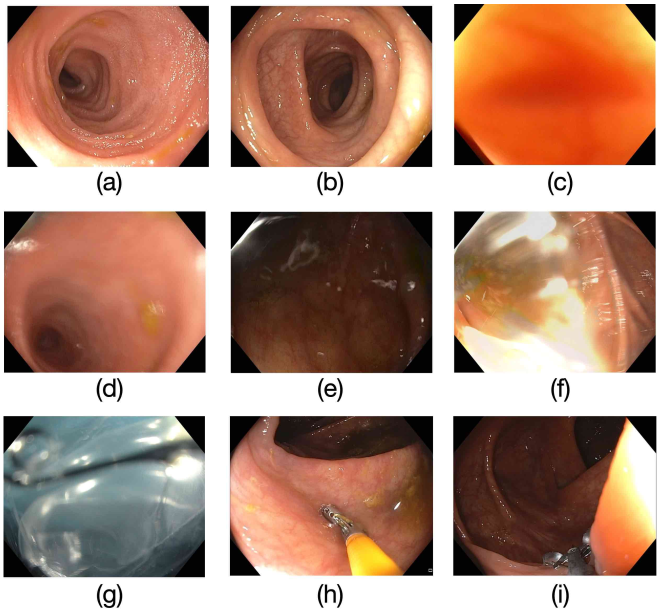


Figure 3.3: Examples of informative frames (a-b); non-informative frames (c-g); and frames with biopsy forceps (h-i). Frames (a) and (b) are examples of informative frames at different colon regions; (c) is a frame captured when the camera was too close to the colon wall; (d) is a frame with significant motion blur; (e) is an underexposed frame; (f) is an overexposed frame; (g) is a frame captured outside of the colon; and (h) and (i) are frames in which biopsies were performed.

3.3.2 Dataset for frame severity classification

To build the dataset for severity classification on individual frames, two broad-certified gastroenterologists manually examined images from patients with UC and annotated them with a severity score using score definitions from MES [range, 0-3]. After that, an adjudicated score was given based on the two reviewers' annotations. Cohen's kappa coefficients were calculated to measure the inter-rater reliability. The Cohen's coefficients for Mayo 0, Mayo 1, Mayo2, and Mayo 3 are 0.650, 0.385, 0.614, 0.750, respectively. The total Cohen's coefficients for the four-class annotation is 0.570. In total, the numbers of images with an adjudicated score of Mayo 0, Mayo 1, Mayo 2 and Mayo 3 in the image severity dataset are 7434 (65.4%), 2211 (19.4%), 1187 (10.4%), 532 (4.7%), respectively. The image classification dataset was randomly split into the training set ($n = 8756$) and test set ($n = 2608$). The training set was used for hyper-parameter tuning and model training. The test set was used to evaluate the performance of the trained models.

3.3.3 Dataset for the localization system

3.3.3.1 Internal Localization Dataset

The localization dataset consists of 44 colonoscopy videos. The videos were collected as described in 3.3.1. The localization dataset was divided into three subsets. Sixteen videos were used to build the camera motion estimation network (Set 1); eighteen colonoscopy videos were used for colon template building (Set 2), and the remaining ten videos, which were paired with ScopeGuide videos, were used as an independent evaluation of the localization algorithm (Set 3). Set 1 was further randomly divided into a training set ($n = 10$), validation set ($n = 3$), and test set ($n = 3$). Each video contains over 3,000 samples, and in total, there are 36,665 samples in the training set. The training and validation set were used for hyper-parameter tuning. The test set was used to evaluate the performance of the trained models. For all videos in the localization dataset, the time point at which the camera was withdrawn was manually annotated by the colonoscopy performer. For all videos in the localization dataset, the camera was withdrawn at the cecum.

3.3.3.2 External EndoSLAM dataset [105]

The dataset provides videos from different cameras on ex-vivo porcine gastrointestinal organs. A robotic arm was used to track the camera trajectory and quantify the six degree-of-freedom pose values. Six videos from three trajectories (Colon-IV Trajectory-2, Small Intestine-IV Trajectory-1, Stomach-II Trajectory-4) were publicly available with the corresponding ground truth for the camera pose. Each trajectory was recorded by a high-resolution endoscopic camera and a low-resolution endoscopic camera.

3.3.4 Dataset for MES estimation

3.3.4.1 Internal MES estimation dataset

The internal MES estimation dataset consists of 51 colonoscopy videos from UM, whose MES were annotated by a broad-certified gastroenterologist. The videos were collected as described in §3.3.1. Videos underwent MES annotation by two local central reviewers blinded to clinical status.

3.3.4.2 External MES estimation dataset

External colonoscopy videos were used to evaluate the proposed MES estimation algorithm, which are from the LYC-30937-EC study, an international phase II randomized clinical trial of an investigational oral therapy for moderate to severe UC (ClinicalTrials.gov identifier NCT02762500). The clinical trial videos were collected from 72 sites (United States, Canada, and 5 European countries). The variation in endoscopic site, equipment, and recording techniques provide an advantage for testing the performance of automated analysis methods in real-world settings. Colonoscopy videos were centrally reviewed for MES by external reviewers as part of the original study protocol and served as the ground truth. The investigators in the presented analysis did not participate in the central review scoring process of external videos from the clinical trial. Clinical trial videos were not used in the development of the proposed colonoscopy-based CDS system.

While the videos in internal MES estimation dataset has a more even distribution of endoscopic severity (MES 0,1 58.8%; MES 2,3 41.2%), the external videos from the clinical trial is more severe (MES 0,1 16.3%; MES 2,3 83.7%, P<0.0001). These differences are unsurprising as clinical trial subject recruitment skews towards more severe disease activity.

Table 3.1 presents the patient characteristics in internal MES estimation dataset

Characteristics	Internal MES dataset (n=51)	External MES dataset (n=124)
Age, years		
mean (SD)	43.5 (15.4)	41.5 (12.8)
Sex, n (%)		
Female	22 (43.1)	52 (41.9)
BMI, kg/m2		
mean (SD)	27.1 (5.5)	25.7 (4.7)
Disease Duration, years		
mean (SD)	8.4 (7.4)	7.7 (6.8)
Total Mayo Score		
mean (SD)	3.9 (2.7)	7.9 (1.6)
C-Reactive Protein, n (%)		
≥5mg/L	n/a †	61 (49.6)
Fecal Calprotectin Range, n (%)		
≤250	n/a †	27 (22.1)
>250 to ≤500	n/a †	19 (15.6)
>500	n/a †	76 (62.3)
Medication Use, n (%)		
None	0 (0.0)	2 (1.6)
5-ASA	34 (66.7)	109 (87.9)
Corticosteroids	8 (15.7)	68 (54.8)
Thiopurines	15 (29.4)	32 (25.8)
Biologic Exposure	18 (35.3)	25 (20.2)
Race, n (%)		
American Indian or Alaskan Native	0 (0.0)	0 (0.0)
Asian	1 (2.0)	1 (0.8)
Black or African American	4 (7.8)	3 (2.4)
Native Hawaiian or Pacific Islander	0 (0.0)	0 (0.0)
White	46 (90.2)	119 (96.0)
Other	0 (0.0)	1 (0.8)
Ethnicity, n (%)		
Hispanic or Latino	1 (2.0)	7 (5.6)

† Prospective C-reactive protein and fecal calprotectin levels were inconsistently available in the developmental video set.

Table 3.1: Patient characteristics in internal MES estimation dataset and external MES estimation dataset. SD: standard deviation

and external MES estimation dataset.

3.3.5 Dataset for outcome prediction

External colonoscopy videos were used to evaluate the predictive value of the spatial severity distribution in outcome prediction, which are from the UNIFI study [107] (ClinicalTrials.gov identifier NCT02407236), a phase III trial of ustekinumab that involved patients with moderate-to-severe UC (defined as a total score [range, 0 to 12] of 6 to 12 on the Mayo scale and an MES [range, 0 to 3] of 2 or 3) [68, 108]. The total Mayo score [108] is comprised of 4 parts: stool frequency, rectal bleeding, endoscopic findings (i.e., MES) and physician's global assessment, each scored from 0-3. The characteristics of patient enrolled in UNIFI study is presented in [107].

The UNIFI trial included an 8-week randomized induction trial and a 44-week randomized-withdrawal maintenance trial. In the induction trial, patients received ustekinumab or placebo at week 0 and were evaluated at week 8 or 16. A primary end point of the UNIFI trial is clinical remission, defined as a total score of ≤ 2 on the Mayo scale [range, 0 to 12] and no subscore > 1 [range, 0 to 3] on any of the four Mayo scale components.

In this study, colonoscopy videos collected at week 0 and week 8 (or 16) were used as input data. The patient's clinical remission at week 44 was used as the patient's outcome. For the outcome prediction model's development and validation, patients didn't complete the trial, patients received a treatment switch, and patients with colonoscopy videos of very poor quality were removed. In total, the outcome prediction dataset consists of 356 patients, and 227 of them had a positive outcome at week 44.

3.4 Methods

3.4.1 Image classification

We proposed an image classification workflow that can be used for non-informative frame classification, severity classification, and biopsy forceps detection.

Pre-processing was first performed on frames sampled from the colonoscopy videos or UC images for standardization. The frames were binarized to identify the largest 4-connected component. After that, the smallest bounding box containing the largest component was used to crop the image. Zero paddings were added to fill the rectangular region into a square region, after which the image was resized to 256 × 256. These resized images were then inputted into the CNN.

In this study, the Inception-v3 architecture [109] was used for non-informative frame classification. Non-informative frame detection and removal can reduce computational load and avoid unexpected errors in camera localization. Similarly, another DL model with Inception-v3 architecture was built to detect frames with biopsy forceps. During a biopsy, the scene between consecutive frames may not be static, but the colonoscope stays at a similar location. In this work, we identified frames during a biopsy by detecting frames with biopsy forceps. After frames with biopsy forceps were detected, frames at 1 second before or after the detected frame were regarded as frames where a biopsy exists. Removing those frames can avoid inaccurate camera motion estimation from the non-rigid scene. The choice of 1 second was made empirically by watching the colonoscopy videos. The Inception-v3 architecture is a 42-layer CNN. It was chosen for image classification in this study because it has achieved success in multiple visual tasks [109]. Considering the large number of parameters for Inception-v3, in the training phase the networks were initialized using a pre-trained model on ImageNet [110]. For biopsy forceps detection, only the last fully-connected layer was fine-tuned using the training set. L2 regularization and dropout were used to improve the model's generalizability.

3.4.2 Camera motion estimation

3.4.2.1 Preprocessing

In this study, the camera at the tip of the colonoscope has a fisheye lens (Olympus PCF-H190). Camera calibration was performed to estimate the camera's intrinsic matrix. With the mathematical model of a fisheye camera proposed in [111], the distorted images from colonoscopy videos were corrected. Details of the camera model and image distortion correction are covered in Appendix B.

3.4.2.2 Architecture

After the camera calibration and image distortion correction, the corrected frames were used for camera motion estimation. An overview of the camera motion estimation architecture is depicted in Figure 3.4. Let us denote a corrected frame from a colonoscopy video with size $H \times W \times C$ at time point t as $I_t : \Omega_I \to [0, 1]$, where $\Omega_I = \{1, 2, \dots, H\} \times \{1, 2, \dots, W\} \times \{1, 2, \dots, C\}$. The unit of time is arbitrary, with a smaller time duration from t to $t+1$ allowing for better estimation. In this study, the duration from t to $t+1$ was chosen to be $\frac{1}{15}$ second. Given a pair of consecutive frames I_t, I_{t+1}, the dense optical flow can be calculated using PWC-Net [112] (the details of which are discussed in Appendix C), which achieved the highest accuracy on several published datasets. Optical flow is a way to describe the magnitude and orientation of apparent velocities of brightness within an image. The dense optical flow from I_t to I_{t+1} is denoted as $F_{t\to t+1} : \Omega_F \to \mathbb{R}$, where $\Omega_F = \{1, 2, \dots, H\} \times \{1, 2, \dots, W\} \times \{1, 2\}$. The pattern in optical flow $F_{t\to t+1}$ shows how a rigid scene changes with the camera's motion from t to $t + 1$. After the optical flow calculation, a concatenation of I_t, I_{t+1}, $F_{t+1\to t}$ is fed into the motion network to estimate the 6 degree-of-freedom of the camera's motion: $\hat{s}_{t\to t+1} = [\hat{t}_x, \hat{t}_y, \hat{t}_z, \hat{r}_x, \hat{r}_y, \hat{r}_z] \in \mathbb{R}^6$. Simultaneously, I_t is fed into the

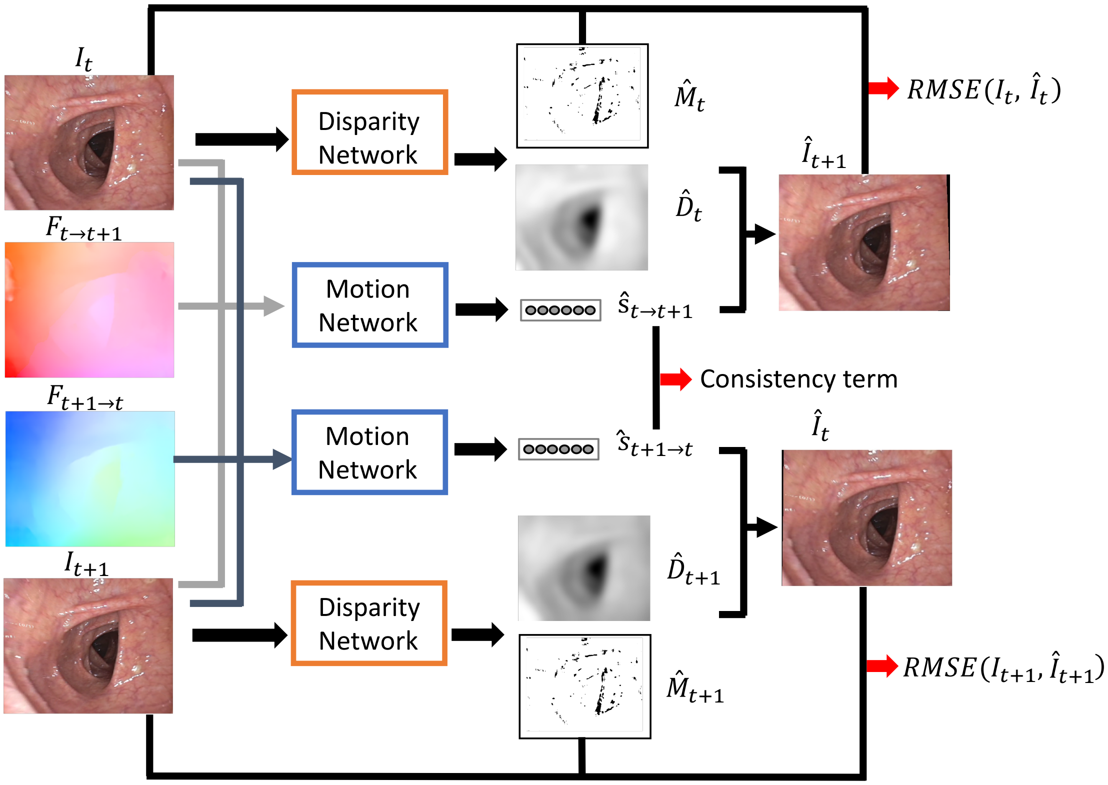


Figure 3.4: Camera motion estimation network. The input to the network is a pair of original frames I_t, I_{t+1} and corresponding optical flows $F_{t \to t+1}, F_{t+1 \to t}$ (visualized using color-coding). The network consists of two sub-networks: the disparity network and the motion network. Loss sources are given following the red arrows.

disparity network to estimate the corresponding disparity map $\hat{D}_t : \Omega_D \to [0, 1]$, where $\Omega_D = \{1, 2, \dots, H\} \times \{1, 2, \dots, W\}$, and the specular region mask $\hat{P}_t : \Omega_P \to [0, 1]$, where $\Omega_P = \{1, 2, \dots, H\} \times \{1, 2, \dots, W\}$. In this study, a value in the disparity map is defined as the inverse of the corresponding scene depth. Similarly, the optical flow $F_{t+1 \to t}$ from I_{t+1} to I_t is calculated, and $\hat{s}_{t+1 \to t}$, $\hat{D}_{t+1}$, $\hat{P}_{t+1}$ are estimated by the motion network and disparity network, respectively. In this study, the estimated disparity map was only used to facilitate the loss calculation. After model training, only the motion network was used for camera localization.

Figure D.1 in Appendix D shows the detailed structure of the motion network. In the motion network, the input is either $\{I_t;\ I_{t+1};\ F_{t \to t+1}\}$ or $\{I_{t+1};\ I_t;\ F_{t+1 \to t}\}$.

While the optical flow pattern can provide information about the camera's motion, the original frames can provide more information about the scene structure, reducing ambiguity in motion detection. The motion network contains eight convolutional layers. The first seven convolutional layers have a filter size of 3×3 and are followed by a max-pooling layer and a ReLU activation function. The last convolutional layer has six filters of size 1×1. The first three filters in the last convolutional layer are used to estimate the predicted camera translation in the x-, y-, and z-axes, while the last three filters are used to estimate Euler angles of the predicted camera rotation. The last convolutional layer is followed by a global average pooling layer to aggregate predictions at all spatial locations. Using the max-pooling layers, the motion network can capture both global and local optical flow patterns for camera motion estimation.

Figure D.2 in Appendix D shows the detailed structure of the disparity network. The input of the disparity network is a corrected frame. Previous literature has shown that non-linear transformations can be modeled to convert a single view image to its corresponding disparity map [113, 114]. The disparity network has an encoder-decoder structure. Unlike in previous literature, the disparity network performs two tasks: disparity estimation and specular region estimation. Considering that both tasks involve intensity and low-level textural feature analysis, a multi-task strategy is applied here, where the two tasks share the same encoder and then have two individual decoders with similar architectures. The encoder consists of three convolutional layers and two max-pooling layers, while the decoder consists of three convolutional layers and two up-sampling layers. All convolutional layers have a filter size of 3×3. For disparity estimation, the last convolutional layer is followed by a sigmoid activation function and a multiplication with 10 to constrain every entry in the estimated disparity map to $[0, 10]$. For specular region estimation, the last convolutional layer is followed by a softmax activation function. All other convolutional layers are followed by a ReLU activation function. The encoder-decoder structure facilitates local and global information extraction and integration.

3.4.2.3 Loss function

(A) Regular photometric loss

For camera motion estimation, the photometric error between the synthesized new frame and the real frame is used as the loss function. Given a frame I_t, estimated disparity map $\hat{D}_t$, and estimated camera motion $\hat{s}_{t\rightarrow t+1}$, a new frame is synthesized at time point $t+1$ by applying differentiable image warping.

Let p_t denote the coordinate of one pixel in the image plane at time point t. If we assume the world frame and the camera frame at time point t are the same, from equations (B.1) and (B.2), the homogeneous pixel coordinates can be projected back to the 3D world coordinate as:

$$p_w = \hat{D}_t(p_t)^{-1}K^{-1}p_t, \tag{3.1}$$

where $p_t = [x, y, 1]^\intercal$, $x \in [1, 2, \ldots, H], y \in [1, 2, \ldots, W]$, are the homogeneous pixel coordinates in I_t, and $p_w = [X, Y, Z]^\intercal, X \in \mathbb{R}, Y \in \mathbb{R}$, and $Z \in \mathbb{R}$, are the 3D world coordinates of the object shown at p_t.

After the camera's motion $\hat{s}_{t\rightarrow t+1}$, p_t will be projected onto the new image plane

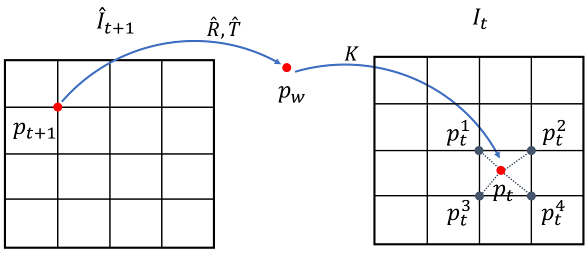


Figure 3.5: A diagram of the image warping process. To synthesize $\hat{I}_{t+1}$, the coordinate p_{t+1} in $\hat{I}_{t+1}$ can be projected back to the world coordinate frame as p_w, from which p_t in I_t can be calculated. p_w is the world coordinate of an object, and p_t, p_{t+1} are the coordinates of this object shown in I_t, I_{t+1}, respectively. We assume that the intensity value $I_{t+1}(p_{t+1})$ is equal to $I_t(p_t)$. The intensity value at p_t can be approximated using bilinear interpolation with its four neighbors.

as:

$$p_{t+1} = K(\hat{R}p_w + \hat{T}) \tag{3.2a}$$

$$\hat{R} = R_x(\hat{r}_x)R_y(\hat{r}_y)R_z(\hat{r}_z) \tag{3.2b}$$

$$R_x(\hat{r}_x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\hat{r}_x & -sin\hat{r}_x \\ 0 & sin\hat{r}_x & \cos\hat{r}_x \end{pmatrix} \tag{3.2c}$$

$$R_y(\hat{r}_y) = \begin{pmatrix} \cos\hat{r}_y & 0 & \sin\hat{r}_y \\ 0 & 1 & 0 \\ -\sin\hat{r}_y & 0 & \cos\hat{r}_y \end{pmatrix} \tag{3.2d}$$

$$R_z(\hat{r}_z) = \begin{pmatrix} \cos\hat{r}_z & -\sin\hat{r}_z & 0 \\ \sin\hat{r}_z & \cos\hat{r}_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.2e}$$

$$\hat{T} = \begin{pmatrix} \hat{t}_x \\ \hat{t}_y \\ \hat{t}_z \end{pmatrix} \tag{3.2f}$$

From equations (3.1) and (3.2), all pixel coordinates in I_t can be projected back into the world coordinates, and then projected to the estimated image plane I_{t+1}, and vice versa. An image warping process is used to generate the new frame $\hat{I}_{t+1}$. Figure 3.5 presents a diagram of the image warping process. Given a pixel coordinate p_{t+1} from $\hat{I}_{t+1}$, p_t, the corresponding coordinates of this object at I_t can be calculated using equations (3.1) and (3.2). The intensity value $I_{t+1}(p_{t+1})$ is the same as $I_t(p_t)$ based on two assumptions: that the colon surface exhibits Lambertian reflectance and the camera motion between consecutive frames is very small. As shown in Figure 3.5, to calculate $\hat{I}_{t+1}(p_{t+1})$, the estimated intensity at the position p_{t+1} in the new frame, bilinear interpolation is used to approximate $I_t(p_t)$ using its four pixel neighbors p_t^1, p_t^2, p_t^3, p_t^4 as

$$\hat{I}_{t+1}(p_{t+1}) = I_t(p_t) = \sum_{i=1}^{4} w_i I_t(p_t^i), \tag{3.3}$$

where w_i is the relative spatial distance between p_t and p_t^i, and

$$\sum_{i=1}^{4} w_i = 1. \tag{3.4}$$

We denote the warping process from I_t to $\hat{I}_{t+1}$ as $\mathcal{W}_{t\to t+1}$, thus we have $\mathcal{W}_{t\to t+1}(I_t) = \hat{I}_{t+1}$

With the image warping process, $\hat{I}_{t+1}$ is synthesized. Similarly, $\hat{I}_t$ can be synthesized given $\hat{s}_{t+1\to t}$, I_{t+1}, and $\hat{D}_{t+1}$. The photometric loss for the pair I_t, I_{t+1} can be calculated as

$$loss_p = \frac{1}{Z}\sum_{p\in\Omega_I} \|I_t(p) - \hat{I}_t(p)\|^2 + \frac{1}{Z}\sum_{p\in\Omega_I} \|I_{t+1}(p) - \hat{I}_{t+1}(p)\|^2, \tag{3.5}$$

where $Z = H \cdot W \cdot C$.

(B) Corrected photometric loss

During the calculation of $\hat{I}_{t+1}$, the calculated p_t may be out of the frame I_t, and usually, a value of 0 will be assigned. The photometric differences in those regions should not be used as supervision for motion estimation because they result from missing information. In this study, an intensity of -1 was assigned instead. A mask M_{t+1} can then be built for $\hat{I}_{t+1}$, where $M_{t+1}(p) = 1$ if $\hat{I}_{t+1}(p) = -1$ otherwise $M_{t+1}(p) = 0$. The mask M_{t+1} can be used to filter out invalid pixel locations that were mapped out of the frame I_t.

In the image warping process, we assume the colon surface exhibits Lambertian reflectance, which is not true for specular regions. Figure 3.6 shows two examples in which the brightness of specular regions within I_t and I_{t+1} changes significantly with different angles of view. As such, pixel locations in specular regions should be excluded when the photometric loss is calculated. In our previous work, we attempted to calculate specular region masks by converting the RGB image into HSV space and then extracting the specular region using a threshold on the saturation channel. Figure 3.6 shows examples of threshold-based mask as P_t and P_{t+1}. While this method is simple and can provide information on the specular region, it is very hard to find an optimal threshold that generalizes for all frames and videos. To estimate the specular region more accurately, one should also consider intensity statistics and textural features. As such, a branch of the disparity network was used for specular region estimation. As shown in Figure D.2, another decoder is used to estimate the specular region mask, and a softmax activation function is applied to the feature map from the last layer to generate a probabilistic map. Examples of the output $\hat{P}_t$, $\hat{P}_{t+1}$ are shown in Figure 3.6, where a smaller value indicates the location is more likely to be within a specular region. With estimated $\hat{s}_{t\to t+1}$, $\hat{D}_{t+1}$, the same image warping process $\mathcal{W}_{t\to t+1}$ will be applied to $\hat{P}_t$ to generate the specular region mask for $\hat{I}_{t+1}$.

Based on the estimated masks $\hat{P}_t$, $\hat{P}_{t+1}$, and M_t, M_{t+1}, a corrected photometric loss can be written as:

$$\begin{aligned} loss_{cp} = \frac{1}{Z}\sum_{p\in\Omega_I} M_t(p) * \|\hat{P}_t(p) * I_t(p) - \mathcal{W}_{t+1\to t}(\hat{P}_{t+1})(p) * \hat{I}_t(p)\|^2 + \\ \frac{1}{Z}\sum_{p\in\Omega_I} M_{t+1}(p) * \|\hat{P}_{t+1}(p) I_{t+1}(p) - \mathcal{W}_{t\to t+1}(\hat{P}_t)(p) * \hat{I}_{t+1}(p)\|^2, \end{aligned} \tag{3.6}$$

where $*$ denotes element-wise multiplication.

$$\begin{aligned} loss_{cp} = \frac{1}{Z}\sum_{p\in\Omega_I} M_t(p) * \|S_t(p) * I_t(p) - \hat{S}_t(p) * \hat{I}_t(p)\|^2 + \\ \frac{1}{Z}\sum_{p\in\Omega_I} M_{t+1}(p) * \|S_{t+1}(p) * I_{t+1}(p) - \hat{S}_{t+1}(p) * \hat{I}_{t+1}(p)\|^2, \end{aligned} \tag{3.7}$$

An additional cross-entropy loss was calculated to train the model for specular region estimation:

$$loss_{ce} = cross\_entropy(P_t, \hat{P}_t) + cross\_entropy(P_{t+1}, \hat{P}_{t+1}), \tag{3.8}$$

where P_t is calculated using a threshold of 0.1 determined by visual evaluation. $loss_{ce}$ is proposed as a weak supervision for specular region estimation.

A combination of $loss_{ce}$ and $loss_{cp}$ encourages the network to detect the specular region and also minimize the photometric error. Figure 3.6 shows examples of $\hat{P}_t$ and $\hat{P}_{t+1}$. The disparity network detects more comprehensive specular regions as compared to the threshold-based method.

The last row of Figure 3.6 depicts the benefits of calculating $loss_{cp}$. The movements of the edge segment in (a) and vessel pattern in (b) should be the primary cues used to estimate the camera's motion. However, the photometric difference between I_{t+1} and $\hat{I}_{t+1}$ around the specular region is very high, which may overwhelm the photometric difference from other regions. As a result, with $loss_p$, the network will be encouraged to reduce the photometric difference on specular regions with a higher priority. Considering the specular regions are not Lambertian surfaces, the

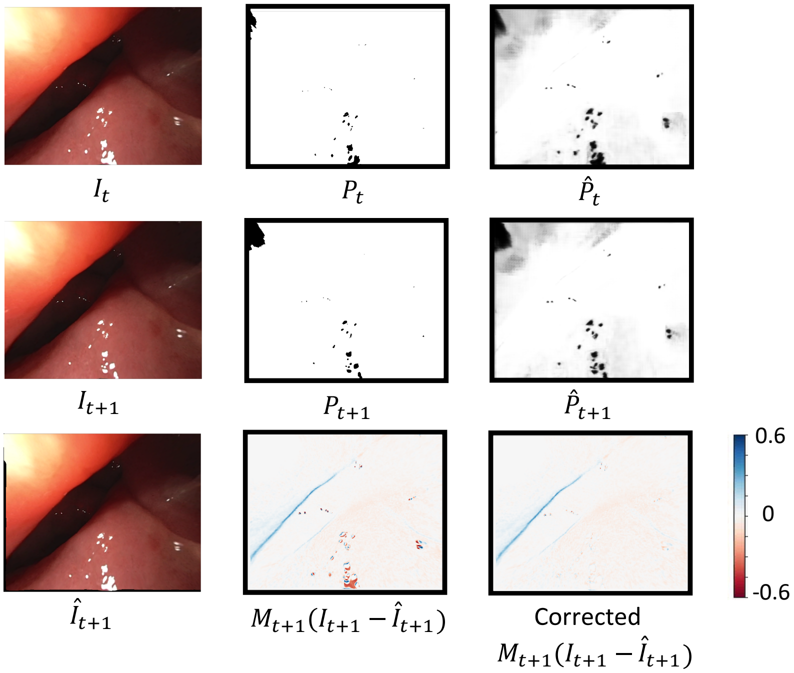


Figure 3.6: An example pair of consecutive frames on specular mask estimation. Threshold-based specular mask (middle column) and the specular mask estimated from the disparity network (right column) are presented. The photometric error map between the projected image $\hat{I}_{t+1}$ and I_{t+1} without and with the estimated specular mask are shown.

photometric difference on specular regions may be inaccurate. The problem can be fixed by applying the estimated specular mask to correct the photometric loss.

(C) Movement consistency loss

The forward movement $\hat{s}_{t\to t+1}$ and backward movement $\hat{s}_{t+1\to t}$ are estimated by the motion network. Let us denote the transformation matrix from frame t to $t+1$ as $Q_{t\to t+1} \in \mathbb{SE}(3)$ and denote its inverse as $Q_{t+1\to t}$, then

$$Q_{t\to t+1} \times Q_{t+1\to t} = Q_{t\to t} = I, \tag{3.9}$$

where × denotes matrix multiplication and I denotes the identity matrix.

As there is no camera movement from frame t to itself, $Q_{t\to t}$ is an identity transformation in homogeneous coordinates.

A movement consistency loss term can be written as

$$loss_{mc} = \|\hat{Q}_{t\to t+1} - \hat{Q}^{-1}_{t+1\to t}\|_F, \tag{3.10a}$$

$$\hat{Q}_{t\to t+1} = \begin{pmatrix} \hat{R}_{t\to t+1} & \hat{T}_{t\to t+1} \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix}, \tag{3.10b}$$

$$\hat{Q}_{t+1\to t} = \begin{pmatrix} \hat{R}_{t+1\to t} & \hat{T}_{t+1\to t} \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix}, \tag{3.10c}$$

$$\hat{Q}^{-1}_{t+1\to t} = \begin{pmatrix} \hat{R}^T_{t+1\to t} & -\hat{R}^T_{t+1\to t} \times T_{t+1\to t} \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix}. \tag{3.10d}$$

The movement consistency term encourages the network to output a forward movement $\hat{s}_{t\to t+1}$ and backward movement $\hat{s}_{t+1\to t}$ that satisfy equation (3.9).The movement consistency term was added into the final loss to improve the network's generalizability.

(D) Final loss

The final loss function can be written as:

$$loss = loss_{cp} + \lambda_{ce} loss_{ce} + \lambda_{mc} loss_{mc} + \lambda_{smo} loss_{smo}, \tag{3.11}$$

where $loss_{smo}$ is from previous literature [88] on monocular depth estimation and is used to encourage the estimated disparity map to be locally smooth. λ_{ce}, λ_{mc}, and λ_{smo} are loss weights.

3.4.3 Camera trajectory and location index estimation

The camera motion network was applied to colonoscopy video in the withdrawal phase to enable successive estimation of the camera's motion, along with the coor-

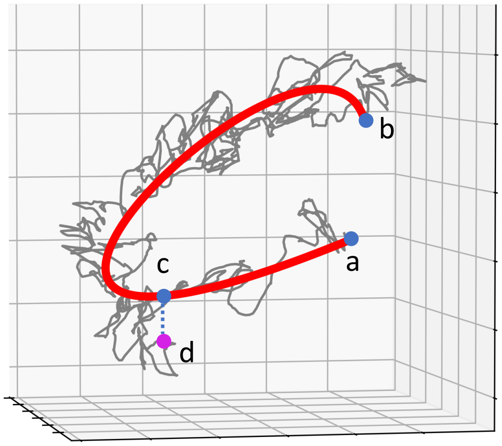


Figure 3.7: A diagram of the location index estimation. The gray line is a camera trajectory and the red line is the major traveling path. a, b are the start and end coordinates in the major traveling path of the camera, respectively. d is the coordinate of the frame at the time point t in the camera trajectory, and c is the closest point on the red line to d.

dinates of the camera (i.e., camera trajectory) after the camera's withdrawal begins. The relative location index is defined from 0 to 1. When the camera starts to be withdrawn at the beginning of the colon (cecum), the frame's location index is 0; when the camera stops at the end of the colon (rectum), the frame's location index is 1. Let l_t denote the colon length that the camera traversed from the beginning to time point t and l_{all} denote the total length of the colon from the cecum to rectum. The location index for the frame at time point t can be calculated as l_t/l_{all}.

The length of the camera trajectory (gray line in Figure 3.7) can not be used directly to estimate the location index as the camera may be moved about to inspect the colon's surface. Considering that the shape of the colon is curved, we propose to estimate a major traveling path for location index calculation. Figure 3.7 depicts how the location index is calculated. The gray line is a camera trajectory in the withdrawal phase. The camera trajectory zigzags because the camera will turn about, moving back and forth to inspect the colon mucosal surface. The red line is the major traveling path of the camera. To estimate the major traveling path of the camera, we used a B-spline curve fitting algorithm with a smoothing factor of γ to fit the original camera trajectory. Let us assume a is the beginning of the colon; b is the end of the colon; and d is the position of the camera at time point t. l_t can be calculated as the length of the red line from a to c, and l_{all} can be calculated as the length of the red line from a to b. Then the location index for the frame at the time point t is the ratio of the length from a to c over the length from a to b in the red line.

The calculation of the frame's location index can convert the frame from the time domain to the location domain. A frame with a location index x can be denoted as I^d_x. Let us denote $f_v : [1, 2, \ldots, N] \to [0, 1]$ as a function that maps the time index of a frame to a location index in the colonoscopy video v, with N being the maximal time index. Given I_t - a frame at the time point t in v - the estimated location index for this frame can be written as $f_v(t)$, and the frame can then be denoted as $I^d_{f_v(t)}$. Figures 3.8(a)-(c) show examples of time-location mapping. If the time domain is used, the frames can be denoted as $I_1, \ldots, I_t, \ldots, I_N$, where $1 \leq t \leq N$; if the location domain is used, the frames can be denoted as $I^d_{f_v(1)}, \ldots, I^d_{f_v(t)}, \ldots, I^d_{f_v(N)}$. If I_a and I_b are the frames at the beginning and end of the colon, respectively, we have $f_v(a) = 0$ and $f_v(b) = 1$.

3.4.4 Anatomical colon segment classification

Anatomical colon segment classification can be performed based on the calculated colon location index. It not only provides contextual information for severity assessment but also helps to validate the performance of the location index estimation. In the proposed anatomical colon segment classification method, we assume that the relative length of each colon segment is similar across patients. The colon segments used in this study include the cecum, ascending colon, transverse colon, descending colon, sigmoid colon, and rectum. In anatomical colon segment classification, we assume that camera withdrawal begins at the cecum.

To build the colon template, the the times at which the camera enters each colon

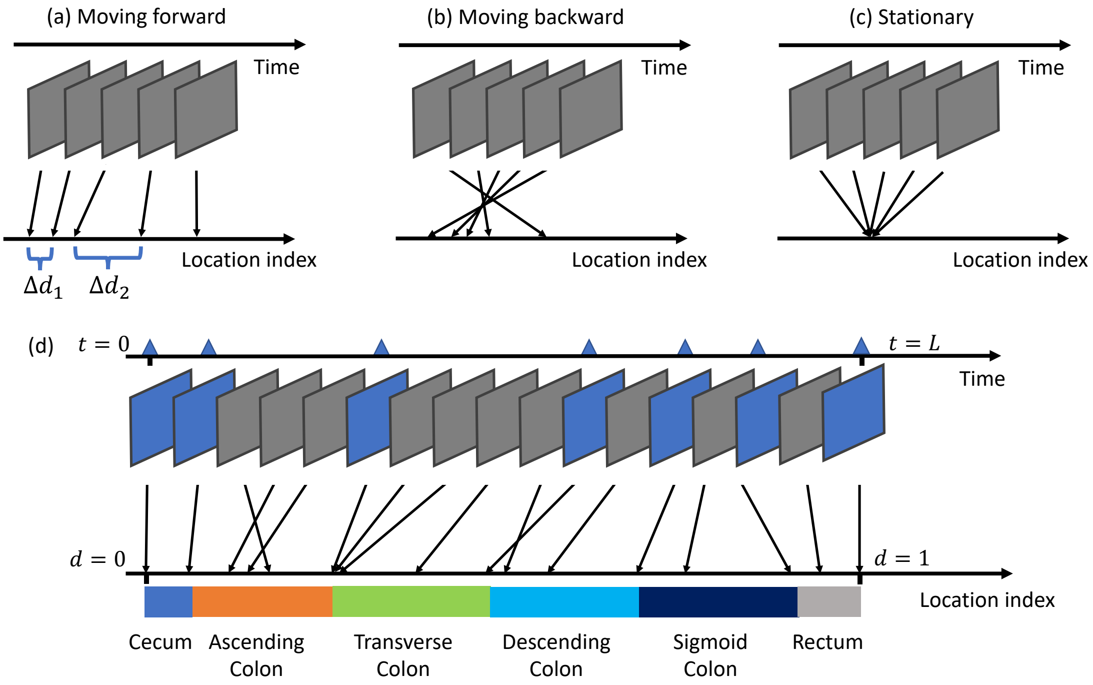


Figure 3.8: A diagram of time-location mapping and colon template building. (a)-(c) are examples when the camera moves forward, backward, or is stationary. Δd_1 and Δd_2 are the same duration in time, but Δd_2 is much larger than Δd_1. (d) shows a full colonoscopy frame sequence in the withdrawal phase mapped into the location domain. To build the template, time points when the colonoscope enters and exits each colon segment were annotated by a physician (blue triangles). Based on the time annotations and time-location mapping, the relative length of each colon segment can be calculated.

segment in the withdrawal phase were annotated. As shown in Figure 3.8 (d), given a colonoscopy video v, a vector q_v^t contains 7 times manually annotated by physicians as

$$q_v^t = [t_v^1, t_v^2, t_v^3, t_v^4, t_v^5, t_v^6, t_v^7], \tag{3.12}$$

where the first 6 entries are the times at which the camera enters the rectum, ascending colon, transverse colon, descending colon, sigmoid colon, and cecum in the withdrawal phase, respectively, and the last entry is the time point when the camera stops. A corresponding vector q_v^s containing the location index of the frames when the camera enters each colon segment and stops at the rectum can be estimated as

$$q_v^s[i] = f_v(q_v^t[i]), \tag{3.13}$$

where $q_v^s[i]$ is the i^{th} element of q_v^s, $1 \leq i \leq 7$, and $f_v(q_v^t[1]) = 0$ and $f_v(q_v^t[7]) = 1$.

The relative length of each colon segment in the colonoscopy video v can be recorded in the vector q_v^{rl} as

$$q_v^{rl}[i] = f_v(q_v^t[i+1]) - f_v(q_v^t[i]), \tag{3.14}$$

where $q_v^{rl}[i]$ is the i^{th} element of q_v^{rl}, and $1 \leq i \leq 6$.

The colon template can be estimated by annotating the times at which the camera enters each colon segment for N colonoscopy videos $v_1, v_2, \ldots, v_N$. The colon template q^{ct} can be estimated as

$$q^{ct}[i] = \frac{w}{N} \sum_{j=1}^{N} q_{v_j}^{rl}[i], \tag{3.15}$$

where $1 \leq i \leq 6$, and w is a scaling factor and

$$\sum_{i=1}^{6} \frac{w}{N} \sum_{j=1}^{N} q_{v_j}^{rl}[i] = 1. \tag{3.16}$$

With a new colonoscopy video, the start of the camera's withdrawal can be identified using the physician's notes, while the end of the camera's withdrawal can be identified as the time of the last informative frame. Location index estimation is then performed for a frame sequence during the withdrawal phase. By comparing these with the constructed colon template, frames falling within each colon segment can be classified.

3.4.5 MES estimation

For each video, the percentages of informative frames classified as Mayo 0, 1, 2, or 3 were calculated. The MES was inferred based on the proportion of frames in a video for each given MES class (e.g. Mayo 3 comprises 12% of frames, Mayo 2 comprises 25% of frames, Mayo 1 comprises 23% of frames and Mayo 0 comprises 40% of frames).

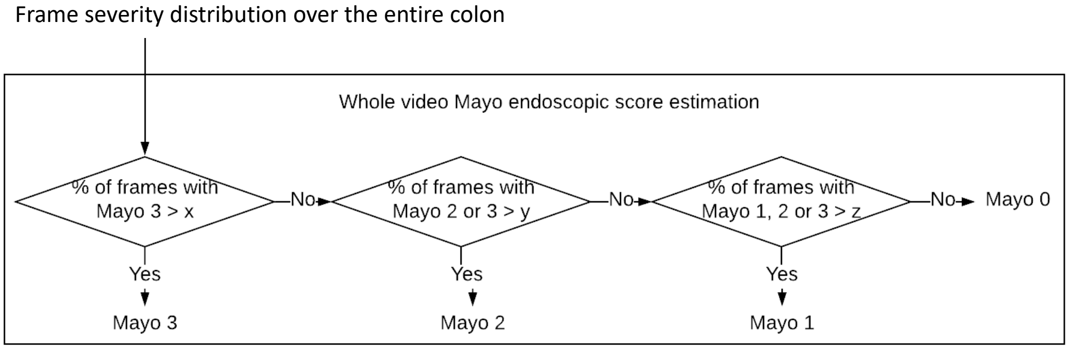

Frame severity distribution over the entire colon
Whole video Mayo endoscopic score estimation
% of frames with Mayo 3 > x
No
% of frames with Mayo 2 or 3 > y
No
% of frames with Mayo 1, 2 or 3 > z
No
Mayo 0
Yes
Mayo 3
Yes
Mayo 2
Yes
Mayo 1

Figure 3.9: A diagram of the proposed MES estimation process.

The highest Mayo score meeting the threshold proportion of frames in a video was selected as the overall Mayo score (as shown in Figure 3.9). The MES proportion thresholds for the overall summary score were determined using a template-matching grid search where the threshold proportions of MES scores in a video were matched to the overall Mayo score provided by an expert review of the entire video. The rationale for requiring a threshold number of video frames to validate the presence of a severity class is to address potential mis-classifications in single-frame severity grading or confounding from other causes that could impact overall scoring. It also corresponds to the fact that a human reviewer does not consider single frames in isolation but actually many seconds worth of video to determine the severity present.

3.4.6 Outcome prediction

In this work, colonoscopy videos collected at week 8 or 16 were used to predict the patient's outcome (clinical remission) at week 44. To extract features from a colonoscopy video, estimated relative location index and severity of individual frames were used to derive the spatial severity distribution over the entire colon. Based on the relative location index, we sampled frames with a step size of 0.001, which means 1000 frames uniformly distributed over the entire colon were sampled. In addition, the colon template from §3.4.4 was used to estimate the anatomical colon segment for

Category	Feature list
Proposed set 1	Age, Sex, Average severity score of the entire colon
Proposed set 2	Age, Sex, Average severity score of the entire colon, Average severity score of individual colon segments
Proposed set 3	Age, Sex, Average severity score of the entire colon, Average severity score of individual colon segments, Annotated MES, Annotated total score
Baseline set 1	Age, Sex, Annotated MES
Baseline set 2	Age, Sex, Annotated MES, Annotated total score

Table 3.2: Feature sets for outcome prediction.

individual frames. Average severity score from the entire colon and average severity scores of individual colon segments (6 colon segments in total: cecum, ascending colon, transverse colon, descending colon, sigmoid colon, and rectum) using the sampled frames were calculated to build the feature representation of a colonoscopy video. With the computed feature representation and patient demographic data (age and sex), a logistic regression model was built for outcome prediction. In addition, two baseline outcome prediction models were built using logistic regression for comparison, where humanly annotated MES and total score were used as predictive variables. The feature sets investigated in this study are described in Table 3.2.

3.4.7 Model training and hyper-parameter tuning

The image classification and motion estimation models were implemented in Tensorflow v1.10 and were trained on an NVIDIA Tesla V100. Adaptive moment estimation was used for optimization.

For the image classification model, the hyper-parameters, including learning rate, batch size, dropout rate and L2 regularization, were chosen via 5-fold cross-validation on the image classification training set. Based on the classification AUCPR, a learning rate of 10^{-3}, a batch size of 8, a dropout rate of 0.4, and a L2 regularization of 0.0001 were selected to train the network on the whole training set. The trained model was then tested on the test set.

For the motion estimation network, Set 1 of the localization dataset was used to build the camera motion estimation model. As mentioned in §3.3.3.1, Set 1 was further split into the training set ($n = 10$), validation set ($n = 3$), and test set ($n = 3$). Different combinations of hyper-parameters, including learning rate, batch size, training steps, λ_{ce}, λ_{mc}, and λ_{smo} were used to train the network on the training set, with the hyper-parameters that achieved the best corrected photometric loss (as discussed in §3.4.8) on the validation set chosen as optimal. Based on the results from hyper-parameter tuning, the final camera motion estimation network was trained with a learning rate of 0.0001, a batch size of 1, 300,000 training steps, a λ_{ce} of 0.01, a λ_{mc} of 100, and a λ_{smo} of 0.02. After hyper-parameter tuning, the trained model was tested on the test set.

For location index estimation, the smoothing factor γ was tuned using anatomical colon segment classification on Set 2 of the localization dataset. A 5-fold cross-validation was performed and $\gamma = 100$ achieved the best average classification accuracy.

A 5-fold cross-validation was used to evaluate the proposed MES estimation method on the internal cohort. In each round, four rounds were used to train the model, and the remaining fold was used as the unseen test data.

3.4.8 Evaluation strategy

3.4.8.1 Classification and prediction tasks

The performance of the classification tasks including non-informative frame classification, biopsy detection, frame severity estimation, MES estimation, and outcome prediction were evaluated using AUPRC, AUC, sensitivity, specificity, precision, and accuracy. For MES estimation, confusion metrics between the estimated Mayo score and annotated Mayo score were calculated. For outcome prediction, a 5-fold cross-validation was used to evaluate the model's performance.

3.4.8.2 Camera motion estimation

To evaluate the performance of the camera motion estimation model, a corrected photometric error (CPE) was calculated as:

$$\begin{aligned} CPE = \frac{1}{Z}\sum_{p\in\Omega_I} M_t(p) * \|P_t(p) * I_t(p) - \mathcal{W}_{t+1\to t}(P_{t+1})(p) * \hat{I}_t(p)\|^2 + \\ \frac{1}{Z}\sum_{p\in\Omega_I} M_{t+1}(p) * \|P_{t+1}(p)I_{t+1}(p) - \mathcal{W}_{t\to t+1}(P_t)(p) * \hat{I}_{t+1}(p)\|^2, \end{aligned} \tag{3.17}$$

for each pair of frames. The average CPE throughout a colonoscopy video was then calculated, with a lower average CPE indicating that the model has a better capacity for camera motion estimation. Please note that the threshold-based specular mask was applied in CPE calculation for fairness.

For the EndoSLAM dataset, the ground truth of the camera's pose was provided. Absolute trajectory error (ATE), relative pose error on translation (RPE Trans.), and rotation (RPE Rot.) can be calculated [115]. ATE measures the distance between the estimated trajectory and the ground truth trajectory. RPE measures the local accuracy of the trajectory over a pair of consecutive frames. The calculation of those metrics can be found in [115]. As the camera motion was estimated with an arbitrary scale, the two trajectories should first be aligned by finding a similarity transformation S[116]. Lower ATE, RPE Trans., and RPE Rot. indicate better motion estimation.

3.4.8.3 Relative location index estimation

The evaluation of the location index estimation is challenging as there is no ground truth for the location index. In lieu of this, we propose two ways to evaluate the location index estimation.

The first method is to compare the anatomical colon segment classification with baselines. Two baselines were considered for anatomical colon segment classification. For the first baseline, the time index is used to build the colon template and the subsequent classification, which assumes that during the withdrawal phase, the proportional amount of time the camera spends within each colon segment is the same for different patients. For a colonoscopy video v, a vector q_v^{rt} contains the relative time in each colon segment and can be calculated as

$$q_v^{rt}[i] = \frac{1}{t_v^7 - t_v^1}(t_v^{i+1} - t_v^i), \tag{3.18}$$

where $q_v^{rt}[i]$ is the i^{th} element of q_v^{rt} and $1 \leq i \leq 6$. A colon template can then be built for anatomical colon segment classification. The assumption here should not hold as the velocity of the camera's movement varies and is affected by individual colon segment disease severity and condition. As a result, an accurate location index estimation method should lead to higher accuracy in anatomical colon segment classification than using the time index directly. The second baseline for estimating position used the approximate colonoscope insertion length measured by ScopeGuide. The limitation of using ScopeGuide length is that although the length of the inserted tube provides an approximate location index, it is a measure of the amount of scope inserted rather than the true distance the camera traveled. An introduction and discussion of ScopeGuide length can be found in Appendix F.

To evaluate the classification performance, we denote the colon segments from cecum to rectum as 1-6. The multi-class classification accuracy, the maximal absolute difference between the predicted class and annotated class, and the average absolute difference between the predicted class and annotated class were calculated for each video. The averaged value and standard deviation among videos were then calculated. For each colonoscopy video in the test set, an individual confusion matrix was calculated, with the entry (i, j) being the percentage of frames in colon segment i

classified as in colon segment j, where $i, j \in$ {Cecum, Ascending Colon, Transverse Colon, Descending Colon, Sigmoid Colon, Rectum}. The individual confusion matrices from colonoscopy videos in the test set were then averaged to build the final confusion matrix of colon segment classification. Additionally, the F1 score, sensitivity, precision, specificity, and accuracy of the individual colon segment classifications were calculated.

The second method for evaluating the location index is to compare the trajectory of the location index from camera motion estimation and that from ScopeGuide length. As the ScopeGuide length can accurately indicate the distance that the camera travels when no loops are generated, one should expect a similar pattern in these two trajectories.

3.5 Results and Discussion

3.5.1 Image classification performance

Table 3.3 shows the performance of non-informative frame classification, biopsy forceps detection and frame severity estimation. From the table, non-informative frame classification and biopsy forceps achieved high AUCs and AUCPRs. Both of the trained models will be used to remove non-informative frames and frame capture from biopsy in relative location estimation and Summary MES estimation. The performance of the frame severity estimation is lower because it is quite subjective and also subject to a high inter-rater disagreement.

3.5.2 Camera motion estimation algorithm evaluation

3.5.2.1 Visualization of outputs from camera pose estimation network

Figure 3.10 gives examples of input images, estimated disparity maps, synthesized frames, and error maps from the trained model to illustrate the motion estimation

Model	AUCPR	AUC	F1	Sensitivity	Specificity	Precision
Non-informative frame classification	0.889 (0.058)	0.959 (0.014)	0.824 (0.095)	0.801 (0.084)	0.956 (0.026)	0.732 (0.064)
Biopsy forceps detection	0.942 (0.048)	0.988 (0.010)	0.848 (0.077)	0.917 (0.076)	0.983 (0.015)	0.801 (0.117)
Frame severity estimation	0.824	0.902	0.720	0.757	0.934	0.754

Table 3.3: Performance of image classification models. Average value and standard deviation across colonoscopy videos are provided for non-informative frame classification and biopsy forceps detection. It is not applied to frame severity classification because few images in frame severity dataset are from the same colonoscopy videos.

algorithm. Figure 3.10 (a) presents an example of input frames with thin textural feature. Figure 3.10 (b) presents an example of input frames with many specular reflections.

I_t was fed into the disparity network to generate $\hat{D}_t$. Comparing I_t and $\hat{D}_t$ in Figure 3.10, one can observe the coordinates in the colon lumen were estimated with a lower disparity value, which is consistent with the fact that they are farther from the camera. In contrast, the coordinates in the colon wall, which are close to the camera, have a disparity value near 1. One can also observe that the specular regions impair the disparity map as those regions are very bright; as such, there is no information to infer the disparity.

With I_t, $\hat{D}_t$ and $\hat{s}_{t \to t+1}$, the frame at $t+1$ can be synthesized as $\hat{I}_{t+1}$. With an accurate estimation, $\hat{I}_{t+1}$ should be closer to the real frame I_{t+1} than I_t. In Figure 3.10, the difference map between the original frames I_t and I_{t+1}, $Error(I_t, I_{t+1})$, is compared with the difference map between $\hat{I}_{t+1}$ and I_{t+1}: $Error(\hat{I}_{t+1}, I_{t+1})$. We can observe that the absolute photometric difference around the edges is quite high in $Error(I_t, I_{t+1})$ because of the camera's motion; those values are largely reduced in $Error(\hat{I}_{t+1}, I_{t+1})$. In addition, one can observe that the absolute photometric difference in specular regions is high in both $Error(I_t, I_{t+1})$, and $Error(\hat{I}_{t+1}, I_{t+1})$, which indicates the necessity of using specular masks for loss calculation.

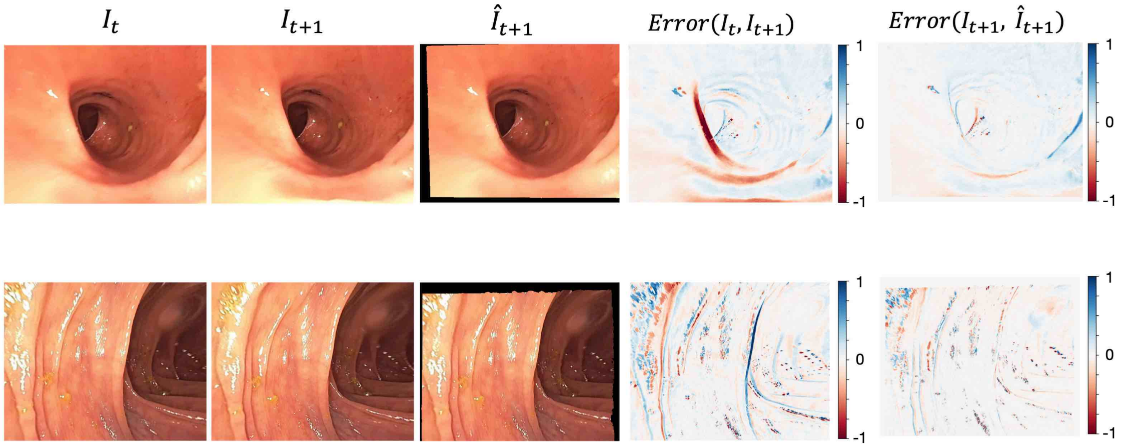


Figure 3.10: Examples of estimated disparity maps, synthesized frames, and corresponding errors. I_t and I_{t+1} are the pair of input fed into the motion estimation network; $\hat{D}_t$ is the estimated disparity map for I_t; and $\hat{I}_{t+1}$ is the synthesised frame given I_t, $\hat{D}_t$, and $\hat{s}_{t\to t+1}$. $Error(I_t, I_{t+1})$ shows the photometric difference map between I_t and I_{t+1}. $Error(\hat{I}_{t+1}, I_{t+1})$ shows the difference map between the synthesised frame and the real frame.

3.5.2.2 Evaluation of the proposed movement consistency term

The training and validation sets of the localization dataset Set 1 were used to find the optimal value for λ_{mc}. From our experimental results, $\lambda_{mc} = 100$ leads to the best motion estimation performance. To further explore the consistency issue, models trained with $\lambda \in \{0, 10, 100, 500\}$ were tested on the test set. The visualizations of the "forward" trajectory and "backward" trajectory for two videos are shown in Figure 3.11, where the "forward" trajectory and "backward" trajectory were aligned by a similarity transformation for a better comparison. Each column in Figure 3.11 presents camera trajectories derived from a model trained with different λ_{mc}. As mentioned in §4.2, the pose estimation from the network is of arbitrary scale. The magnitude of $loss_{mc}$ can affect the scale of the estimated pose. As such, for each λ_{mc}, the estimated camera pose was scaled by a factor that constrains the length of the estimated "forward" trajectory for the first video in the test set to 10. The scaling factor was used for a fair comparison among models trained with different λ_{mc} values.

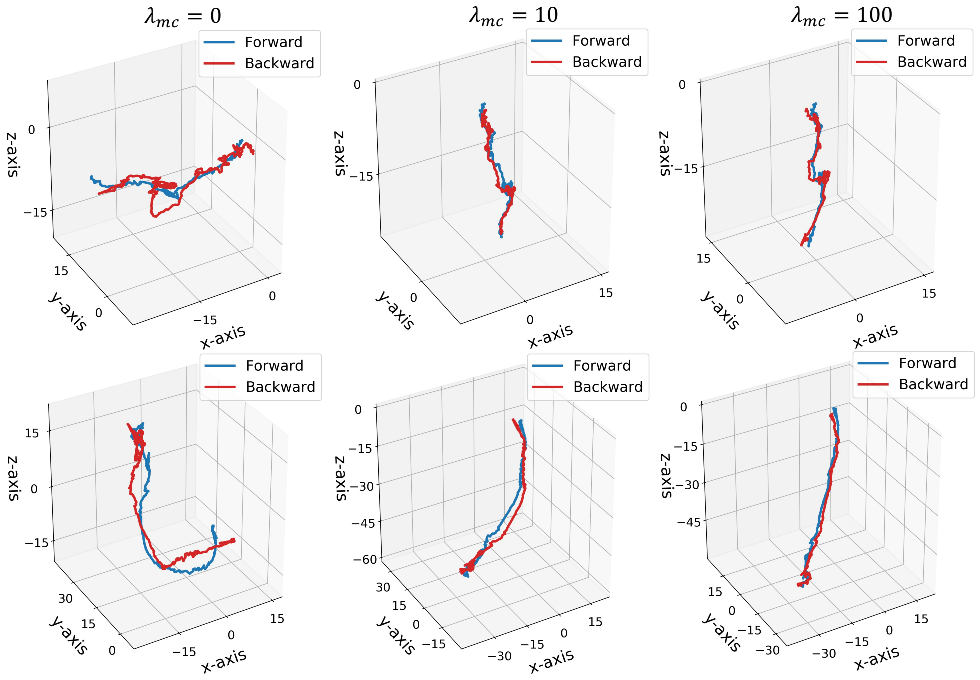


Figure 3.11: The comparison of camera trajectories computed using $\hat{s}_{t\to t+1}$ ("forward") and using $\hat{s}_{t+1\to t}$ ("backward"). Each column presents trajectories from a model trained with a different λ value.

3.5.2.3 Performance of the proposed camera pose estimation method

	Train	Test
Proposed	0.0490 (0.0090)	0.0567 (0.0032)
with fixed specular mask	0.0510 (0.0088)	0.0580 (0.0033)
w/o specular mask	0.0496 (0.0091)	0.0593 (0.0037)
w/o optical flow as input	0.0485 (0.0073)	0.0591 (0.0039)
w/o consistency term	0.0488 (0.0079)	0.0581 (0.0025)

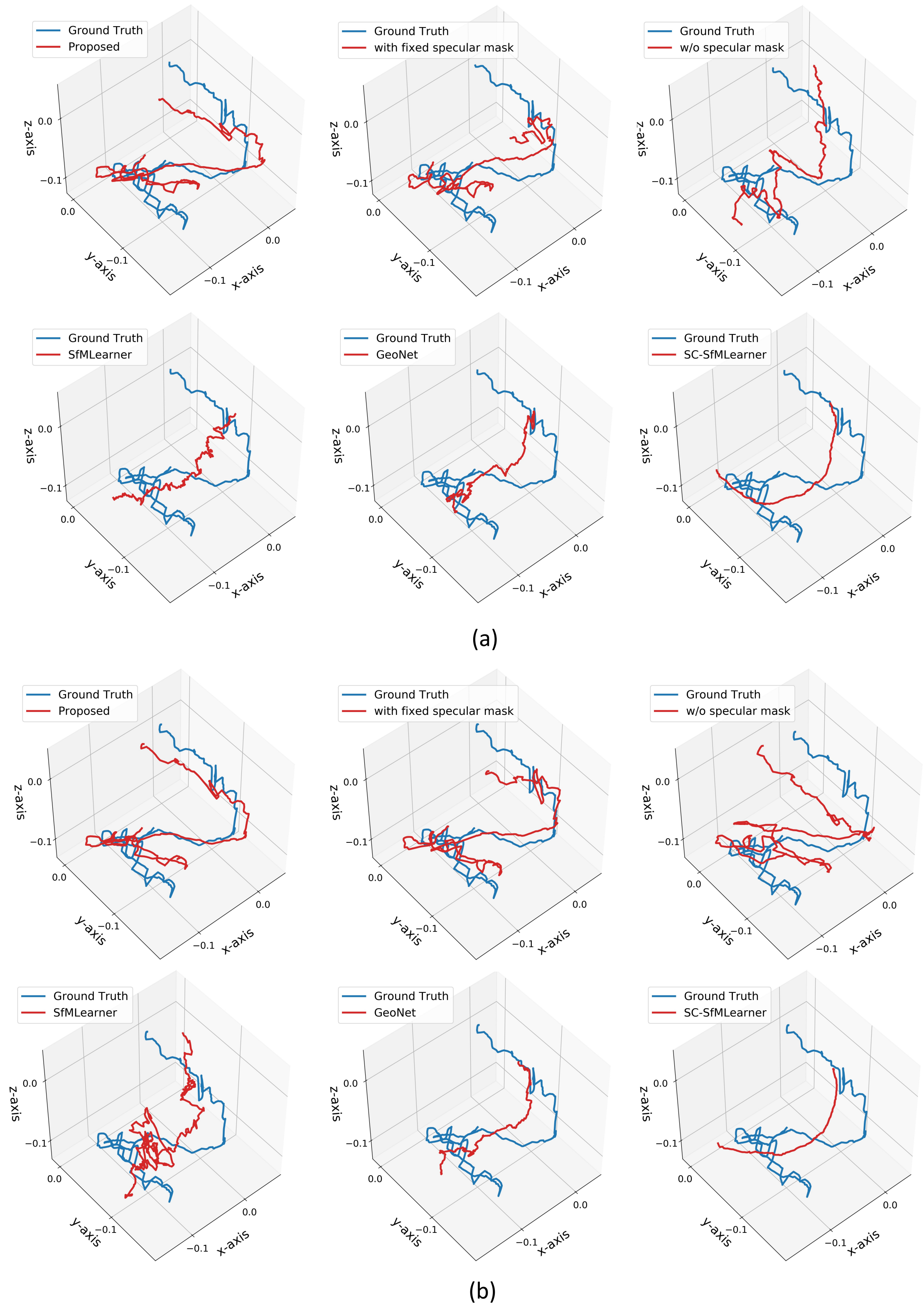
Table 3.4: CPE on the training and test set of localization dataset Set 1.

For the proposed camera pose estimation method, three major modifications were made, including correcting photometric loss with an estimated specular mask, adding optical flow as input, and adding movement consistency terms. In Table 3.4, the proposed model and the proposed method without one of these modifications were evaluated on the training and test set of the localization dataset Set 1. Models presented includes the proposed method ("Proposed"), the proposed method with a corrected photometric loss using a threshold-based specular mask ("with fixed specular mask"), the proposed method without applying the specular mask to correct the photometric loss ("w/o specular mask"), the proposed method without using optical flow as input ("w/o optical flow as input"), and the proposed method without adding movement consistency term ("w/o consistency term").

Further, we evaluated the proposed camera pose estimation on the EndoSLAM dataset. The EndoSLAM dataset is an external dataset with ground truth for camera pose. Besides the aforementioned models, four existing techniques: "SfMLearner"[87], "GeoNet"[93], "SC-SfMLearner"[96], and "optical flow-based" [78] from recent literature with publicly available code were also evaluated on the EndoSLAM dataset for comparison.

Figure 3.12 compares the estimated trajectory and ground truth trajectory on Small Intestine-IV Trajectory-1. We can observe that the estimated trajectories from the proposed method are the most accurate with loops of similar shape as compared to the ground truth trajectories. In addition, the trajectory estimated on the high-resolution video is very close to the trajectory on the low-resolution video, which indicates a good generalizability of the proposed method. The "with fixed specular mask" model also achieved good performance. However, compared with its estimated trajectory on low-resolution video, the accuracy of the estimated trajectory on high-resolution video decreases.

Quantitative measurements were calculated and are shown in Tables 3.5 and 3.6. Table 3.5 presents the performance of models on individual high-resolution videos. Table 3.6 summarizes the average performance on high-resolution videos, low-resolution videos, and the entire dataset. The proposed method achieved the lowest ATE and RPE Rot. The RPE Trans. values for all methods are quite close. While the "optical flow-based" model has the lowest RPE Trans., its ATE and RPE Rot. are very high. From our experimental results, the proposed method achieves the best performance



(a)

(b)

Figure 3.12: Comparison of the estimated trajectories and ground truth trajectories. (a) Trajectories for the high-resolution video on Small Intestine-IV Trajectory-1. (b) Trajectories for the low-resolution video on Small Intestine-IV Trajectory-1.

Model	Colon-IV Trajectory-2			Small Intestine-IV Trajectory-1		
	ATE	RPE Trans.	RPE Rot.	ATE	RPE Trans.	RPE Rot.
	$(\times 1e^{-2})$	$(\times 1e^{-2})$	$(\times 1e^{-2}\,^{\circ})$	$(\times 1e^{-2})$	$(\times 1e^{-2})$	$(\times 1e^{-2}\,^{\circ})$
Proposed	**1.98 (0.61)**	**0.16 (0.16)**	**0.85 (0.83)**	**2.82 (1.14)**	0.22 (0.15)	**0.83 (0.74)**
w/o specular mask	2.98 (1.27)	0.20 (0.21)	2.93 (3.89)	5.77 (2.17)	0.22 (0.18)	3.76 (4.32)
w/o consistency term	3.01 (1.20)	0.20 (0.17)	0.98 (0.84)	4.77 (1.88)	**0.20 (0.09)**	0.96 (0.75)
w/o optical flow as input	2.99 (1.31)	**0.16 (0.16)**	1.03 (0.85)	3.52 (1.66)	0.22 (0.15)	1.09 (0.73)
SfMLearner [87]	2.98 (1.15)	0.24 (0.20)	1.40 (0.97)	5.09 (1.97)	0.21 (0.14)	1.05 (0.81)
GeoNet [93]	2.75 (1.08)	0.18 (0.16)	1.32 (0.89)	4.31 (1.77)	0.21 (0.12)	1.38 (0.83)
SC-SfMLearner [96]	2.96 (1.08)	0.21 (0.19)	1.08 (0.79)	4.23 (2.11)	0.21 (0.16)	0.94 (0.70)
optical flow-based [78]	3.15 (1.38)	0.19 (0.19)	4.97 (4.98)	4.89 (2.43)	**0.20 (0.15)**	5.34 (4.27)

Table 3.5: Performance comparison of motion estimation algorithms on high-resolution videos from EndoSLAM dataset. The standard deviation (std) was calculated over pairs of the consecutive frames. The results are presented in the format of mean (std). "Proposed": the proposed method; "with fixed specular mask": the proposed method with a corrected photometric loss using a threshold-based specular mask; "w/o specular mask": the proposed method without applying the specular mask to correct the photometric loss; "w/o optical flow as input": the proposed method without using optical flow as input; "w/o consistency term": the proposed method without adding the movement consistency term.

Model	High-resolution videos			Low-resolution videos		
	ATE	RPE Trans.	RPE Rot.	ATE	RPE Trans.	RPE Rot.
	$(\times 1e^{-2})$	$(\times 1e^{-2})$	$(\times 1e^{-2}\,^{\circ})$	$(\times 1e^{-2})$	$(\times 1e^{-2})$	$(\times 1e^{-2}\,^{\circ})$
Proposed	**2.48 (0.36)**	0.24 (0.07)	**0.86 (0.03)**	**2.36 (0.30)**	0.34 (0.12)	**1.10 (0.24)**
with fixed specular mask	3.22 (0.71)	**0.23 (0.07)**	0.97 (0.01)	2.44 (0.64)	0.34 (0.12)	1.13 (0.22)
w/o specular mask	3.44 (0.34)	0.24 (0.07)	1.06 (0.02)	3.14 (0.94)	0.34 (0.12)	1.29 (0.25)
w/o optical flow as input	4.01 (0.74)	0.24 (0.05)	0.96 (0.02)	3.55 (0.76)	0.34 (0.11)	1.17 (0.24)
w/o consistency term	4.29 (1.15)	0.26 (0.06)	3.76 (0.67)	3.25 (0.99)	0.34 (0.11)	3.22 (1.77)
SfMLearner [87]	4.12 (0.87)	0.27 (0.07)	1.58 (0.52)	3.98 (0.97)	0.36 (0.12)	2.99 (0.46)
GeoNet [93]	3.77 (0.72)	0.24 (0.07)	1.35 (0.02)	3.56 (0.95)	0.34 (0.11)	1.25 (0.28)
SC-SfMLearner [96]	3.81 (0.60)	0.25 (0.06)	1.02 (0.06)	3.78 (0.92)	0.35 (0.10)	1.09 (0.16)
optical flow-based [78]	4.07 (0.72)	**0.23 (0.05)**	5.47 (0.47)	4.05 (0.75)	**0.33 (0.08)**	6.11 (0.95)

Table 3.6: Performance comparison of motion estimation algorithms on the EndoSLAM dataset. The standard deviation (std) was calculated over videos. The results are presented in the format of mean (std).

on the EndoSLAM dataset.

3.5.3 Camera trajectory comparison

With the trained camera pose estimation model, the camera's trajectory can be derived and the relative location index can be calculated. Figure 3.13 compares the location index calculated from camera motion estimation (orange line), ScopeGuide length (blue line), and frame index (dotted gray line) of the 10 colonoscopy videos in Set 3 of the localization dataset. From Figure 3.13, we can observe that the ScopeGuide length-based location index is rougher due to its low resolution. Also, sharp spikes or bumps can be observed. In those spikes (shown in green boxes), the location index may increase or decrease by 0.2 (usually equal to 15-25 cm) in a short time. A physician manually examined the paired colonoscopy videos and ScopeGuide videos, finding that those spikes or bumps occur because of loops generated when the colonoscopy performer advanced the colonoscope toward the cecum. In Figure 3.13 (i), a sharp increase of the location index can be observed, which results from the loop generated in the insertion phase. Those spikes and bumps can significantly impair the proposed colon segment classification when the location index is derived from the ScopeGuide length, where the max segment error is 1.9 (0.3). Except for regions with shape spikes or bumps, the blue line and orange line have good consistency in the pattern of the location index sequence. It indicates that using a computer vision-based method can lead to a good sense of the camera's location. By comparing the dotted gray line with the other two lines, we can find sampling frames using the time index can lead to an unbalanced sampling of frames at different colon regions, emphasizing the importance of camera localization in colonoscopy video analysis.

3.5.4 Anatomical colon segment classification performance

When comparing the performance of the colon segment classification using motion-based location index and using the time index, Set 2 of the localization dataset was used for template building and Set 3 of the localization dataset was used for testing.

The colon template built from manual time annotations and the location index from our localization system is shown in the first row of Table 3.7, wherein each entry corresponds to the relative length of each colon segment (from cecum to rectum). From the estimated colon template, the cecum and rectum are shorter, occupying less than 10% of the total colon length, while the other four colon segments are longer, occupying 15% to 20% of the total colon length. The template is consistent with our physiological knowledge about the colon. We then performed colon segment

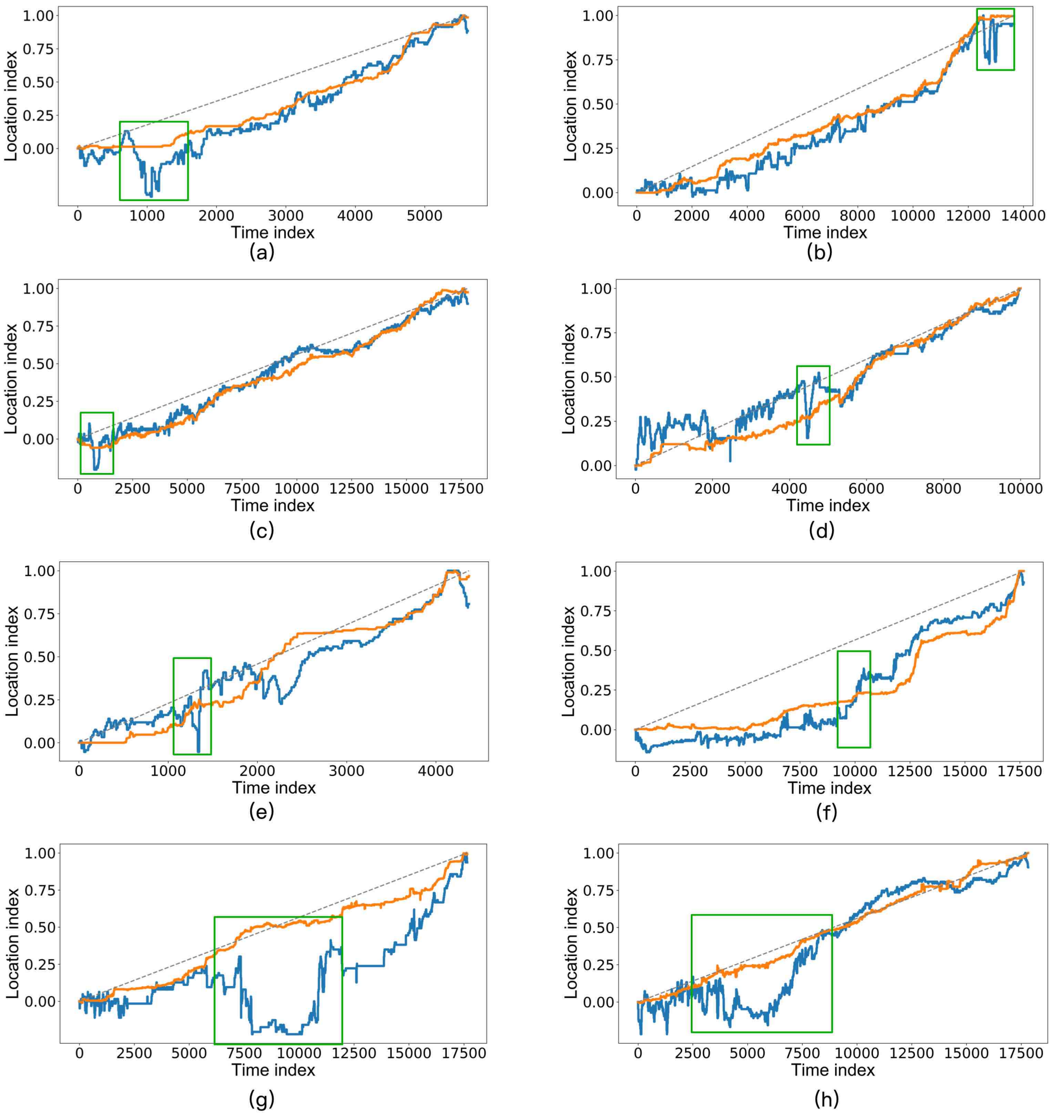


Figure 3.13: Trajectories comparisons of location index calculated from different sources using Set 3 of the localization dataset. Orange line: location index derived from the camera motion estimation; Blue line: location index calculated using length in ScopeGuide videos; Dotted gray line: location index using the time index directly; Green box: regions with sharp increase/decrease in ScopeGuide length result from generated loops.

classification on the test set.

The average accuracy and per-segment errors of the classification on the test set are given in Table 3.8. The confusion matrix of the classification is shown in Figure 3.14 (a). The classification performance on individual colon segments is shown in Table 3.9. From the results, the anatomical colon segment classification is most accurate in the cecum and rectum and less accurate in the middle colon segments. This is

Method	Cecum	Ascending Colon	Transverse Colon	Descending Colon	Sigmoid Colon	Rectum
Motion-based location index (Set 2)	0.061	0.146	0.224	0.223	0.258	0.088
Time index (Set 2)	0.068	0.217	0.210	0.177	0.172	0.156
Motion-based location index (Set 3)	0.067	0.142	0.245	0.204	0.244	0.096
ScopeGuide Length-based location index (Set 3)	0.236	0.129	0.162	0.204	0.167	0.102

Table 3.7: Colon templates estimated using different strategies. In the first two rows, the colon templates were built on Set 2 of the localization dataset. In the last two rows, the colon templates were the average of those from cross-validation on Set 3 of the localization dataset.

	Accuracy	Average segment error	Max. segment error
Motion-based location index (Set 2)	0.754 (0.111)	0.246 (0.111)	1.0 (0.0)
Time index (Set 2)	0.608 (0.140)	0.399 (0.151)	1.2 (0.4)
Motion-based location index (Set 3)	0.718 (0.097)	0.282 (0.097)	1.0 (0.0)
ScopeGuide Length-based location index (Set 3)	0.587 (0.134)	0.472 (0.172)	1.9 (0.3)
Motion-based location index (Set 2) with all frames	0.576 (0.117)	0.437 (0.131)	1.2 (0.4)
Motion-based location index (Set 3) with all frames	0.579 (0.175)	0.436 (0.189)	1.2 (0.4)

Table 3.8: Classification accuracy and segment errors. In the first two rows and the last two rows, the colon templates were built on Set 2 of the localization dataset. In the middle two rows, the colon templates were the average of those from cross-validation on Set 3 of the localization dataset. In the last two rows, the camera trajectories were estimated using all frames from colonoscopy videos, meaning non-informative frames and frames with biopsy forceps were not removed.

an intrinsic property of the proposed classification method because the classification of the middle colon segments suffers from accumulated error. Also, one can observe that the frame will only be mis-classified to the colon segments adjacent to the true segment.

For comparison, a colon template was built, and colon segment classification was performed using the time index. The colon template is shown in the second row of Table 3.7. Using the time index, the relative lengths of the descending colon and

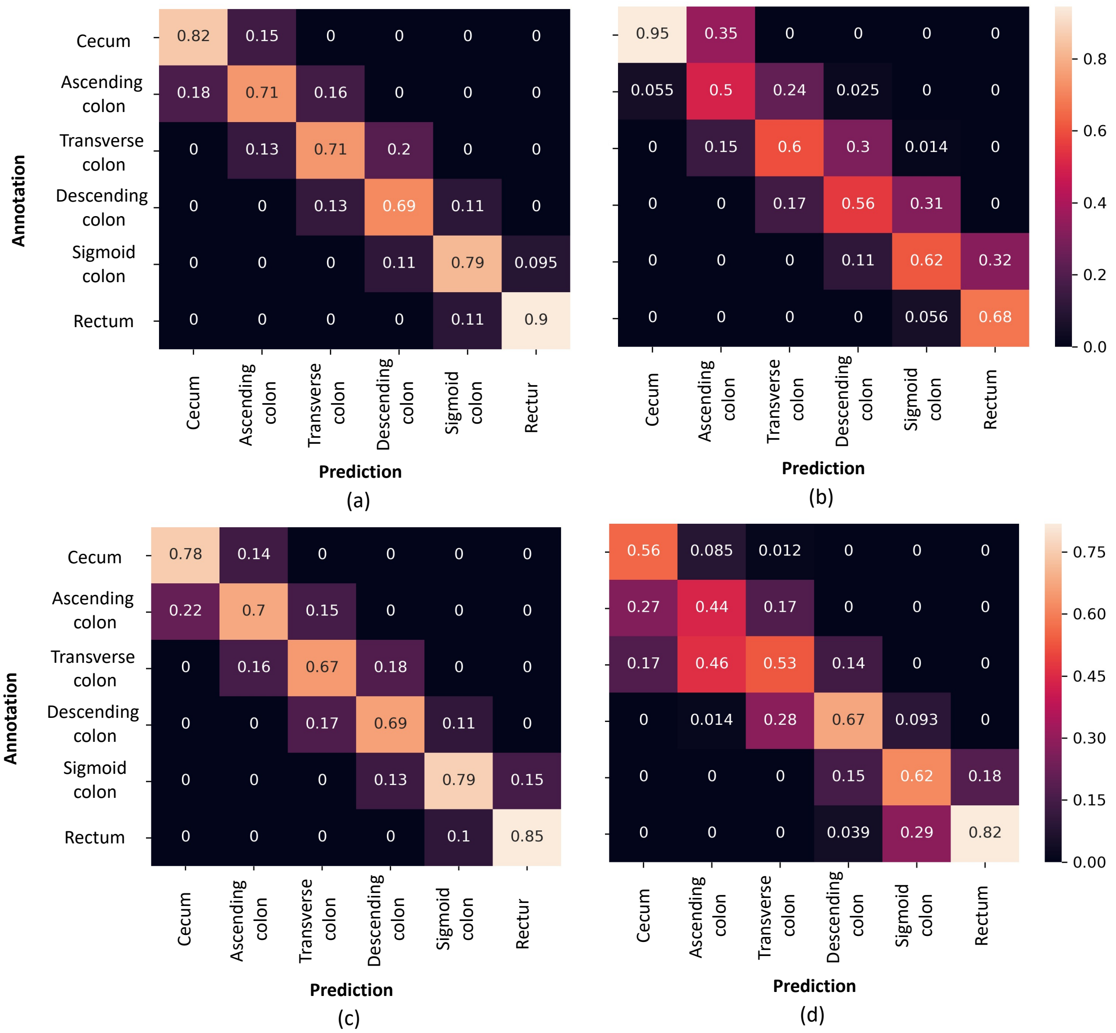


Figure 3.14: (a)-(b): Confusion matrix between the classified colon segment and annotated colon segment on the entire test set. (a) is the result of using the location index, and (b) is the result of using the time index. (c)-(d): Confusion matrix between the classified colon segment and annotated colon segment from leave-one-out cross-validation on the test set. (c) is the result of using the location index, and (d) is the result of using length from ScopeGuide videos. The calculation of the confusion matrix is described in §3.4.8.

	Cecum	Ascending Colon	Transverse Colon	Descending Colon	Sigmoid	Rectum
F1	0.953 (0.035)	0.901 (0.067)	0.874 (0.075)	0.878 (0.076)	0.927 (0.043)	0.975 (0.033)
Sensitivity	0.864 (0.239)	0.696 (0.216)	0.693 (0.216)	0.750 (0.229)	0.786 (0.227)	0.888 (0.257)
Specificity	0.975 (0.021)	0.949 (0.048)	0.928 (0.073)	0.915 (0.065)	0.957 (0.056)	0.988 (0.016)
Precision	0.822 (0.171)	0.714 (0.230)	0.713 (0.282)	0.686 (0.187)	0.785 (0.288)	0.905 (0.139)
Accuaracy	0.953 (0.035)	0.901 (0.067)	0.874 (0.075)	0.878 (0.076)	0.927 (0.043)	0.975 (0.033)

Table 3.9: Anatomical colon segment classification performance on Set 3 of the localization dataset using estimated location index.

	Cecum	Ascending Colon	Transverse Colon	Descending Colon	Sigmoid	Rectum
F1	0.919 (0.059)	0.833 (0.114)	0.828 (0.077)	0.833 (0.063)	0.863 (0.038)	0.940 (0.027)
Sensitivity	0.566 (0.237)	0.753 (0.324)	0.566 (0.220)	0.524 (0.211)	0.606 (0.141)	0.962 (0.114)
Specificity	0.996 (0.012)	0.871 (0.074)	0.900 (0.073)	0.905 (0.043)	0.925 (0.055)	0.945 (0.031)
Precision	0.945 (0.164)	0.500 (0.266)	0.598 (0.303)	0.558 (0.217)	0.616 (0.294)	0.675 (0.189)
Accuaracy	0.919 (0.059)	0.833 (0.114)	0.828 (0.077)	0.833 (0.063)	0.863 (0.038)	0.940 (0.027)

Table 3.10: Anatomical colon segment classification performance on Set 3 of the localization dataset using time index.

	Cecum	Ascending Colon	Transverse Colon	Descending Colon	Sigmoid	Rectum
F1	0.940 (0.044)	0.886 (0.069)	0.864 (0.074)	0.870 (0.069)	0.915 (0.046)	0.963 (0.040)
Sensitivity	0.873 (0.239)	0.636 (0.226)	0.709 (0.219)	0.677 (0.225)	0.716 (0.222)	0.890 (0.270)
Specificity	0.959 (0.044)	0.949 (0.046)	0.909 (0.075)	0.920 (0.060)	0.958 (0.056)	0.976 (0.033)
Precision	0.776 (0.231)	0.699 (0.227)	0.672 (0.276)	0.688 (0.197)	0.787 (0.292)	0.847 (0.206)
Accuaracy	0.940 (0.044)	0.886 (0.069)	0.864 (0.074)	0.870 (0.069)	0.915 (0.046)	0.963 (0.040)

Table 3.11: Anatomical colon segment classification performance from leave-one-out cross-validation on Set 3 of the localization dataset using estimated location index derived.

	Cecum	Ascending Colon	Transverse Colon	Descending Colon	Sigmoid	Rectum
F1	0.855 (0.082)	0.821 (0.065)	0.800 (0.103)	0.876 (0.066)	0.889 (0.063)	0.934 (0.048)
Sensitivity	0.893 (0.149)	0.407 (0.338)	0.381 (0.226)	0.647 (0.250)	0.708 (0.284)	0.590 (0.246)
Precision	0.558 (0.302)	0.438 (0.324)	0.533 (0.251)	0.672 (0.249)	0.619 (0.238)	0.818 (0.272)
Specificity	0.856 (0.104)	0.904 (0.071)	0.916 (0.051)	0.921 (0.061)	0.922 (0.042)	0.984 (0.027)
Accuaracy	0.855 (0.082)	0.821 (0.065)	0.800 (0.103)	0.876 (0.066)	0.889 (0.063)	0.934 (0.048)

Table 3.12: Anatomical colon segment classification performance from leave-one-out cross-validation on Set 3 of the localization dataset using length from ScopeGuide videos.

sigmoid colon are similar to that of the rectum, which is not true. The average accuracy and per-segment errors of the classification are presented in Table 3.8. Both errors and their standard deviations are higher. The corresponding confusion matrix is shown in Figure 3.14 (b), and individual colon segment classification performance is shown in Table 3.10. From these results, one can conclude that using the proposed location index is more accurate than using the time index.

We also compared the performance of anatomical colon segment classification using the camera motion-based location index with that using the ScopeGuide length-based location index. As only paired ScopeGuide videos were available in Set 3, leave-one-out cross-validation was used to evaluate the performance. For each fold, only one video was used for testing, and all others were used to build the colon template. The evaluation measurements were then averaged over all folds.

Using the motion-based location index, the average template from cross-validation on Set 1 is close to the template from Set 2, with around ±2% difference. The average accuracy and segment errors of the classification from cross-validation are shown in Table 3.8. Figure 3.14 (c) and Table 3.11 further show the cross-validation performance of using the motion-based location index. The overall classification performance is slightly lower than the performance in Figure 3.14 (a) and Table 3.9.This may be due to the colon template being built from fewer cases.

For colon segment classification using ScopeGuide length, the average accuracy and segment errors are presented in Table 3.8. Figure 3.14 (d) and Table 3.12 further show the performance from cross-validation. From these results, using the ScopeGuide length-based location index yields poor classification performance, much lower than that from the camera motion-based location index. This may be because the loops generated inside the body lead to sharp increases and decreases in length. This issue is discussed further in the next section. From the template built using the ScopeGuide length-based location index, the cecum segment is estimated to be more than 20% of total colon length. The over-estimation of cecum length may result from the fact that loops are more easily generated when the amount of inserted scope is greater. The trajectory of the location index derived from ScopeGuide length is discussed in the next section.

3.5.5 MES estimation

MES grading thresholds for Mayo 1, 2, and 3 scores (x, y, z in Figure 3.9) of 7%, 6%, and 6%, respectively, were used for entire-video score prediction. The automated MES estimation algorithm exhibited very good agreement with gastroenterologist reviewers (κ=0.84) and correctly predicted the MES in 40/51 (78%) of high-resolution internal videos (the confusion matrix is shown in Table 3.15). Unsurprisingly, disagreement was concentrated in mild disease severity classes including Mayo 1, where 5/9 cases were classified as Mayo 2 and 1/9 were classified as Mayo 0. Paired gastroenterologist reviewers agreed on exact MES in 84.3% of cases (κ=0.95), similarly with disagreement concentrated in the intermediate Mayo 1 and 2 classes.

Agreement between the predicted MES and reference MES provided by external central review was moderate (κ=0.59) with 57.1% (151/264) of videos being correctly graded based on the provided central review score. Automated endoscopic analysis

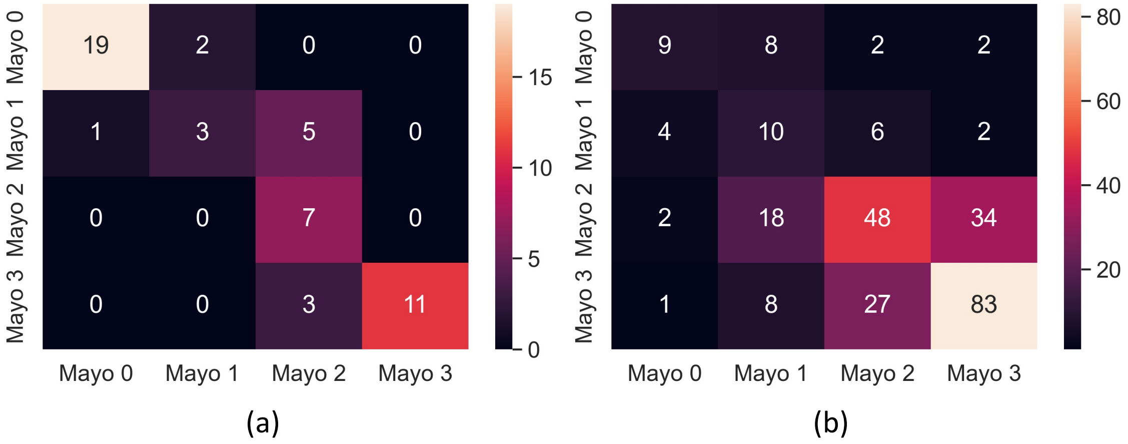


Figure 3.15: Confusion matrix of the MES estimation. (a) Internal MES estimation dataset; (b) External MES estimation dataset.

was within 1 MES severity level of the score provided by central reviewers in 93.5% (247/264) of videos. Fully automated methods correctly separated Mayo 0-1 vs. Mayo 2-3 endoscopic severity in 83.7% (221/264) videos compared to the reference central reviewer score. Qualitative mis-classification analysis of the 17/264 (6.4%) automated predicted MES that were 2 levels different than central review scores was performed. Over-estimated disease severity (e.g. Mayo 0 predicted as 2 or 3) contained extensive biopsy sampling with resulting mucosal bleeding, which was interpreted as severe disease. Under-estimated scores (e.g. Mayo 2 predicted as 0) had short segments of severe disease qualifying the subject as a high endoscopic severity grade, despite the severe disease comprising a small fraction of the disease burden.

3.5.6 Outcome prediction

The performance of the logistic regression-based outcome prediction models are shown in Table 3.13. For the performance of two baseline models, the humanly annotated MES can help predict the patient's outcome, and clinical components (stool frequency, rectal bleeding, and physician rating of disease activity) in total score also add more predictive value. As expected, features extracted from the spatial severity distribution can better characterize the patient's condition. With the average severity

Feature set	Accuracy	Recall	Specificity	Precision	F1	AUC
Proposed set 1	0.681 (0.059)	0.906 (0.072)	0.282 (0.036)	0.689 (0.029)	0.737 (0.045)	0.686 (0.081)
Proposed set 2	0.694 (0.039)	0.891 (0.077)	0.346 (0.031)	0.706 (0.010)	0.741 (0.037)	0.721 (0.074)
Proposed set 3	0.731 (0.068)	0.862 (0.057)	0.500 (0.137)	0.756 (0.058)	0.758 (0.049)	0.792 (0.040)
Baseline set 1	0.611 (0.045)	0.775 (0.082)	0.321 (0.073)	0.668 (0.024)	0.670 (0.042)	0.644 (0.085)
Baseline set 2	0.676 (0.013)	0.848 (0.035)	0.372 (0.036)	0.705 (0.006)	0.723 (0.014)	0.711 (0.029)

Table 3.13: Performance comparison of the outcome prediction models with different features sets. Average and standard deviation from a 5-fold cross-validation are given.

score over the entire colon, the logistic-regression-based model performed better than the humanly annotated MES. Furthermore, with average severity scores from individual colon segments, the model's accuracy outperformed the model with the total score. In the proposed set 3, we combined the estimated average severity scores from the entire colon and individual colon segments with MES and the total score. With the proposed set 3, the logistic-regression-based model achieved an average AUC of 0.79, which is 8% higher than using the annotated MES and total score alone.

3.5.7 Limitations and future work

One limitation of this study is that the time at which the camera was withdrawn was manually annotated. This is due to the high proportion of non-informative frames during the insertion phase. The information loss resulting from frame removal reduces the accuracy of the camera localization during the insertion phase and makes it challenging to identify the time at which the camera was withdrawn automatically. An image classifier will be trained to detect frames in the ileum or cecum based on textural features in our future work. Combined with the motion estimation results, the time at which the camera was withdrawn will be identified, which can lead to a fully automated localization system. The proposed anatomical colon segment classification is limited by assuming the patient's colon and segments are of a regular length. In our future work, additional information, including surgical history and automatically detected anatomical features (e.g., the appendiceal orifice), will be integrated to identify the colon segment better. The scale-drifting problem is an intrinsic limitation of the self-supervised camera pose estimation network. Our experimental results in Figure 3.13 indicate that the scale-drifting problem is minor in the proposed system. In our future work, we will explore the possibility of integrating other information in the localization system. Though the colon surface is relatively uniform, some anatomical features may be detected in the cecum, transverse colon, and rectum. It is possible to build a method that leverages the information from colon segment templates and anatomical feature detection to refine the location index estimation for a new colonoscopy video.

An important limitation in the MES estimation was the difference in disease severity between the subjects of the internal and external videos, who expectantly contained a higher proportion of moderate-to-severe disease. We believe the disease severity distribution typical of clinical trials is justified by the expectation these populations are likely where video analysis will be first applied. Future development and validation methods will benefit from evenly distributed disease severity datasets to be of the most value in research and clinical care. Another limitation is the ground truth for endoscopic disease severity assessments is inherently subjective. However, there is no perfect reference for endoscopic scoring, and central reading does not completely eliminate bias, disagreement, or variability of ground truth disease severity grading. Increasingly, our ground truth for endoscopic feature evaluation and scoring may need to be reconsidered given the increasing availability of computational methods for more discrete and reproducible image assessment.

At the end of this chapter, preliminary results of the proposed outcome prediction model are presented. For the current outcome prediction model, we only investigated the basic statistics features on the severity scores. Other features such as distance metrics evaluating the severity change from week 0 and week 8 (or 16), histogram features on severity change, and features from wavelet transformation may provide more comprehensive information. In our future work, we will continue building feature representation from the spatial severity distribution over the entire colon. From our experimental results, the clinical information from the total score still adds predictive value. As a result, we will add more clinical variables into the model and try to reduce the manual annotation in outcome prediction. Additionally, other ML algorithms will also be applied to improve the prognostic model.

CHAPTER IV

A Novel Tropical Geometry Based Interpretable Machine Learning Method with An Application in Clinical Decision Support for Patients with Advanced Heart Failure

4.1 Introduction

AI/ML techniques have been increasingly applied to healthcare problems [117]. Previous studies investigated the capability of AI in disease diagnosis, treatment effectiveness prediction, and patient outcome prediction [118, 119, 120, 121]. Several studies have shown that AI performs as well as or better than humans [122]. With a lower cost, AI-based decision support systems have the potential to improve patient management.

Despite tremendous progress in the field of AI/ML-based clinical decision support systems, there are still significant challenges that prevent the widespread use of these methods in sensitive clinical applications. While traditional models such as linear regression models and decision trees provide accessible reasoning, these models are less capable of achieving high performance on complicated clinical problems. In contrast, a wide spectrum of ML models with higher complexity, including families of neural networks and support vector machines (SVM), can yield good metrics on experimental datasets. However, these “black box” models lack transparency and justification of their recommendations, making them much less likely to be trusted in clinical applications. Moreover, many popular ML methods, such as deep learning, utilize a large number of parameters, thus requiring large training and validation datasets to avoid overfitting the data. However, in many clinical applications, collecting large annotated training datasets may be costly or even impossible. As such, there is a clear need for an interpretable ML model that can reliably model data using relatively small training sets. In addition, in healthcare applications, there exist many invaluable heuristics derived from domain knowledge expertise, often in the form of approximate rules that are used by human experts. For example, when caring for patients with end-stage heart failure (HF), cardiologists use their clinical intuitions, paired with transplant guidelines, to identify patients who may benefit from a durable mechanical circulatory support (MCS) device or heart transplantation (HT). In the majority of existing AI/ML models, there is no clear mechanism to leverage such approximate knowledge for model formation or training.

The motivation of this study is to solve the aforementioned limitations in the field of AI. In this study, an interpretable ML algorithm is proposed that can not only produce a transparent classification model but also leverage existing domain knowledge to improve model generalizability and reliability. The proposed network is built upon a fuzzy logic and inference system [123, 124], a type of approximate reasoning method that has been heavily used for multidimensional system modeling [125, 126]. In this study, a network with adaptive fuzzy subspace division and rule discovery was developed. In addition, the input encoding functions and the aggregation operators in classical fuzzy inference networks were reformulated by introducing tropical geometry [127], a piecewise-linear version of conventional algebraic geometry. To validate the proposed methods, two synthetic datasets and one practical application in clinical decision support for patients with advanced HF were investigated to demonstrate the capability and interpretability of the proposed model.

The clinical decision support application used in this study is the differentiation of patients eligible for and most likely to benefit from advanced therapies; such as durable MCS, most commonly a left ventricular assist device (LVAD), or HT; from those too well, too sick, or otherwise ineligible for advanced therapies. HF afflicts 6.5 million Americans 20 and older, with its prevalence projected to increase annually [128, 129]. Treatment of these patients remains limited both by medical therapies and by organ availability. The appropriate delivery of advanced therapies (HT or MCS implantation) to patients with end-stage HF is highly nuanced and requires expertise from advanced HF cardiologists. Due to the high prevalence of HF, the majority of patients are managed by primary care physicians or cardiologists, who lack training in the management of these patients. Thus, there is a need for AI-based tools that can systematically identify patients warranting a referral to an advanced HF cardiologist for consideration of HT or MCS implantation.

Our contributions in this study can be summarized as:

1. A novel end-to-end interpretable fuzzy network is proposed, whose resulting recommendations and predictions would be transparent to users such as clinicians and patients. The model can produce humanly understandable rules, while regularization within the model training process encourages parsimonious rules that could be readily incorporated into clinical practice. Moreover, the extracted rules enable the discovery of new clinical knowledge. The proposed network has been validated using synthetic data with ground truth reasoning and a dataset consisting of patients with HF. The experimental results show that the network has the capability to extract hidden rules from datasets. In addition, the proposed network achieved comparable or better performance than other ML models.

2. Using the proposed algorithm, approximate domain knowledge can be directly incorporated into model training. The existing domain knowledge can improve the model's performance and reduce the need for a large training set, which makes it particularly appropriate for clinical applications. From our experimental results, initializing a network with existing approximate knowledge can significantly improve the model's accuracy.

3. The proposed ML algorithm has been validated with an application of identifying patients with HF eligible for advanced therapies, a highly sensitive application in medicine. From our results, the proposed algorithm achieves a smaller generalization error. The rules from the trained network have been visualized and validated by cardiologists. The developed model can improve care for patients with HF by providing assessments that can be used by general providers without HF expertise.

4.2 Related Work

4.2.1 Interpretable ML models

One of the most popular definitions of interpretability is "the ability to explain or to present in understandable terms to a human" [130]. There are primarily two bodies of work related to model interpretability: post-hoc interpretation and transparency[131].

Post-hoc interpretation methods are dedicated to explaining pre-developed "black box" ML models. For example, the interpretability of a random forest model was investigated by measuring variable importance [132]. [133] proposed Local Interpretable Model-agnostic Explanations, which can explain the individual predictions of any classifier by learning local surrogate models that approximate the predictions from the target "black box" model. In [133], an attribution graph discovers and summarizes crucial neuron associations that contribute to a model's predictions. While post-hoc methods can reveal how powerful models works, they are mostly approximations and have limited capacity in elucidating how to improve a model's interpretability.

In contrast, transparency addresses how a model functions internally and can provide exact explanations. A transparent model has an explainable structure design that enables interpretation. While those models are interpretable, they are usually less accurate than powerful "black box" ML models. The simplest transparent models are linear models, but these may fail whenever the relationships between features and responses are non-linear. The Naïve Bayes classifier calculates the probability for a class depending upon the value of the feature so that the contribution of each feature towards a certain class is evident. Decision trees are another class of transparent models that can capture interactions among different features. However, the structure of the decision tree is quite unstable and highly dependent on feature selection for each split. Generalized additive models are extended linear models that can capture non-linear relationships between individual features (or pairwise interactions) and responses [134]. They have been used in practical applications and exhibit good performance and interpretability [135]. However, they are less capable of modeling in high-dimensional feature interactions. Another type of transparent model is a fuzzy inference model, which models the relationship between features and responses by constructing compositional rules [123]. Fuzzy inference models are designed for problems with inherent imprecision and uncertainty. In fuzzy inference models, knowledge is represented in the format of fuzziness of antecedents, consequents, and relations. As rules closely approximate human logic in decision-making, and fuzziness often exists in practical applications and especially in healthcare, the proposed network in this study is designed to leverage fuzzy logic and inference systems.

4.2.2 Fuzzy inference system

Previous studies have shown that fuzzy inference systems can be used for nonlinear system approximation and rule identification [125, 126]. While decisions produced by conventional AI/ML models are often opaque, hindering knowledge extraction and transfer, fuzzy inference models can extract humanly understandable knowledge from data. Classical fuzzy inference models utilize membership functions such as triangular functions to transform crisp inputs to a membership degree of fuzzy concepts. After that, a set of concepts are aggregated by T-norm and T-conorm operators (aggregation operators) to construct if-then rules, with the crisp output from each rule then transformed into output. min (T-norm) and max (T-conorm) are commonly used operators in fuzzy logic [123, 136]. A wide spectrum of fuzzy inference systems utilize the Takagi-Sugeno (TS) inference model [124], whereby a complete rough partition of the input space is generated and an input-output relation is formed for each subspace. Adaptive Network-based Fuzzy Inference System (ANFIS)[137] is a hybrid of a feed-forward neural network and fuzzy inference system with supervised learning capability that can be used to update the input-output relation in each subspace. ANFIS has been successfully applied in multiple applications [138, 139]. In our previous work [140], an adaptive fuzzy inference network was developed and optimized using a genetic algorithm to identify patients eligible for advanced therapies. From our results, the network achieved good classification performance and provided transparent rules.

However, the designs of the TS model and ANFIS pose challenges in practical complex applications where the number of input variables is relatively large as this results in exponential growth in the number of subspaces (as well as the number of parameters). To handle this problem, a flexible k-d tree [141] and quadtree [142] have been adopted for input space partition, but are limited in that it is more challenging to assign understandable terms to membership functions using these methods. In this study, unlike previous methods, we propose an end-to-end network that will adaptively and iteratively discover subspaces related to each class using gradient-based back-propagation.

4.3 Datasets

4.3.1 Synthetic datasets

Two synthetic datasets were constructed by simulating features with fixed distributions and rules to generate responses. The ground truth rules from the synthetic datasets can be used to assess a method's capability in extracting humanly understandable knowledge from the data and modeling the relationship between inputs and responses. In addition, with ground truth rules, synthetic datasets can be used to assess whether the proposed method can benefit from existing knowledge.

For each dataset, a 10-fold cross-validation was used for performance evaluation. In each iteration, the dataset was randomly split into the training set (64%), validation set (16%), and test set (20%).

4.3.1.1 Synthetic dataset 1

Eight input variables were simulated as: $x_1 \sim \mathcal{N}(0,\, 2)$, $x_2 \sim \mathcal{N}(5,\, 3)$, $x_3 \sim \mathcal{N}(-1,\, 5)$, $x_4 \sim \mathcal{N}(1,\, 2)$, $x_5 \sim \mathcal{N}(-2,\, 1)$, $x_6 \sim$ Bernoulli(0.5), $x_7 \sim \mathcal{N}(0,\, 1)$, $x_8 \sim \mathcal{N}(0,\, 1)$. If any of the following rules apply to one observation, then this observation is positive and otherwise negative:

- Rule A: $x_2 < 3.8$ and $x_3 > -2$ and $x_6 = 1$;
- Rule B: $x_2 > 6.3$ and $x_3 > -2$ and $x_6 = 1$;
- Rule C: $x_1 < 1$ and $x_4 > 2$ and $x_6 = 0$;
- Rule D: $x_3 > 0$ and $x_5 > -1$ and $x_6 = 0$;

- Rule E: $x_1 < 1$ and $x_5 > -1.5$ and $x_6 = 0$.

Additionally, random noise sampled from $\mathcal{N}(0,\, 0.01)$ are added to input variables. From the above rules we can readily observe that the response of one observation doesn't rely on x_7 and x_8. x_7 and x_8 are used as irrelevant variables to assess the model's resilience to redundant features.

4.3.1.2 Synthetic dataset 2

Nine input variables were simulated as: $x_1 \sim \mathcal{N}(0,\, 2)$, $x_2 \sim \mathcal{N}(5,\, 3)$, $x_3 \sim \mathcal{N}(-1,\, 5)$, $x_4 \sim \mathcal{N}(1,\, 2)$, $x_5 \sim \mathcal{N}(-2,\, 1)$, $x_6 \sim \mathcal{N}(-1,\, 4.4)$, $x_7 \sim \mathcal{N}(0,\, 1.2)$, $x_8 \sim \mathcal{N}(0,\, 1)$, $x_9 \sim \mathcal{N}(0,\, 1)$. The sample is positive if $(x_1+0.5x_2+x_3)^2/(1+e^{x_6}+2x_7) < 1$.

Unlike synthetic dataset 1, which is built from rules, a highly non-linear function is used to assign the response. Though such a relationship between input variables and responses rarely exists for clinical applications, this dataset is used to determine if the proposed network can still achieve good performance by approximating the complicated relation as simple rules.

4.3.2 HF dataset

A HF dataset is created to train a classification model that identifies patients eligible for advanced therapies. For this analysis, we focused our analysis on the timing of LVAD implantation and urgent HT as these urgent transplants occur in the order of months and can be predicted based on the time of transplant listing. Two cohorts were used in this study.

4.3.2.1 REVIVAL cohort

The REVIVAL (Registry Evaluation of Vital Information for VADs in Ambulatory Life) registry contains information on 400 patients with advanced systolic HF from

21 US medical centers. As part of the registry, patients were evaluated at up to 6 prespecified time points over a 2-year period and underwent relevant examinations. At each time point, investigators were asked to record whether the participant had been evaluated for HT or LVAD and the result of that evaluation. Death, HT, and durable MCS implantation were study endpoints with no additional follow-up. For purposes of this analysis, study participants were labeled at each time point as appropriate (positive) or not appropriate (negative) for advanced therapies. In total, the cohort contains 96 positive samples from 62 patients, and 1336 negative samples from 339 patients.

4.3.2.2 INTERMACS cohort

The INTERMACS (Interagency Registry for Mechanically Assisted Circulatory Support) registry is a North American registry of adults who received an FDA-approved durable MCS device for the management of advanced HF. The registry includes clinical data on all adults ≥ 19 years of age who received a device at one of 170 active INTERMACS centers. The registry includes information on patient demographics, clinical data before and at the time of MCS implantation, and clinical outcomes up to one-year post-MCS implantation or until HT. For this analysis, data was extracted at the time of LVAD implantation and patients classified as "appropriate for advanced therapies." In total, the cohort contains 7781 positive samples from 7813 patients.

Patients from the two cohorts were combined to form a larger dataset. 23 clinical variables were selected by clinicians and used in this study including heart rate, systolic blood pressure (SYSBP), sodium concentration, albumin concentration, uric acid concentration, total distance walked in 6 minutes (DISTWLK), gait speed during a 15 feet walk test (GTSPDTM), left ventricular dimension in diastole (LVDEM), left ventricular ejection fraction severity score (EF), eight-item Patient Health Question-

	REVIVAL		INTERMACS
	Eligible for	Too Well for	Eligible for
	HT/MCS (n=62)	HT/MCS (n=291)	HT/MCS (n=7781)
Age, years			
mean (SD)	60.8 (9.7)	59.2 (11.8)	57.6 (12.9)
Gender, n (%)			
Female	16 (25.8%)	77 (26.5)	1605 (20.6)
NYHA class, n (%)			
I	0 (0%)	6 (2.1%)	274 (3.5%)
II	6 (9.7%)	96 (33.0%)	45 (0.6%)
IIIA	47 (75.8%)	162 (55.7%)	1361 (17.5%)
IIIB	3 (4.8%)	6 (2.1%)	6133 (78.8%)
IV	6 (9.7%)	21 (7.2%)	0 (0%)
INTERMACS profile			
mean (SD)	5.5 (1.0)	6.1 (0.9)	2.6 (1.0)
Heart rate			88.3 (17.4)
mean (SD)	75.3 (12.2)	75.0 (12.5)	
Systolic blood pressure			
mean (SD)	104.6 (11.6)	110.0 (16.3)	106 (15.7)

Table 4.1: Patient characteristics of the REVIVAL and INTERMACS datasets. NYHA: New York Heart Association. SD: standard deviation.

naire depression scale (PHQ-8) score, mitral regurgitation (MITRGRG), lymphocyte percentage (LYMPH), total cholesterol (TCH), hemoglobin (HGB), age, sex, comorbidity index, glomerular filtration rate (GFR), pulse pressure, treatment with cardiac resynchronization therapy, need for temporary MCS device, treatment with guideline directed medical therapy (GDMT) for heart failure, and peak oxygen consumption during a maximal cardiopulmonary exercise test (pVO2). Note, in this study, EF denotes the ejection fraction severity score, which means a patient with a low ejection fraction has a high EF value.

Patient characteristics of the REVIVAL and INTERMACS datasets are shown in Table 4.1. Patient-wise splitting was performed to construct training, validation, and test sets, the details of which are shown in Table 4.2. From the table 4.1, we can observe that patients in INTERMACS cohort are severer than patients in REVIVAL cohort. As a result, all patients in the INTERMACS cohort are used as training samples and the trained model will be validated on positive samples from REVIVAL

	Training set	Validation set	Test set
Patients in REVIVAL with advanced therapy (n=64)	0%	50%	50%
Patients in REVIVAL w/o advanced therapy (n=339)	80%	10%	10%
Patients in INTERMACS (n=2998)	100%	0%	0%

Table 4.2: Ratio of patients from different groups in training, validation, and test set in one iteration

dataset. With this data split, we can better evaluate the generalizability of the proposed method.

Additionally, to facilitate model training, 5 approximate rules denoting eligibility for advanced therapies were collected from heart failure and transplant cardiologists:

- Rule A: EF is high, and pVO2 is low;
- Rule B: EF is high, and DISTWLK is low;
- Rule C: Age is high, EF is high, and SYSBP is low;
- Rule D: EF is high, and MITRGRG is high;
- Rule E: EF is high, and the GDMT is low;

4.4 Methods

4.4.1 Overview of the proposed work

In this study, a transparent end-to-end network was designed that can discover fuzzy subspaces contributing to each class. Figure 4.1 depicts the proposed network. The proposed network and regular neural network are alike in a layer-by-layer structure but entirely different in mathematical modeling. The proposed network has three major components: encoding module, rule module, and inference module. In

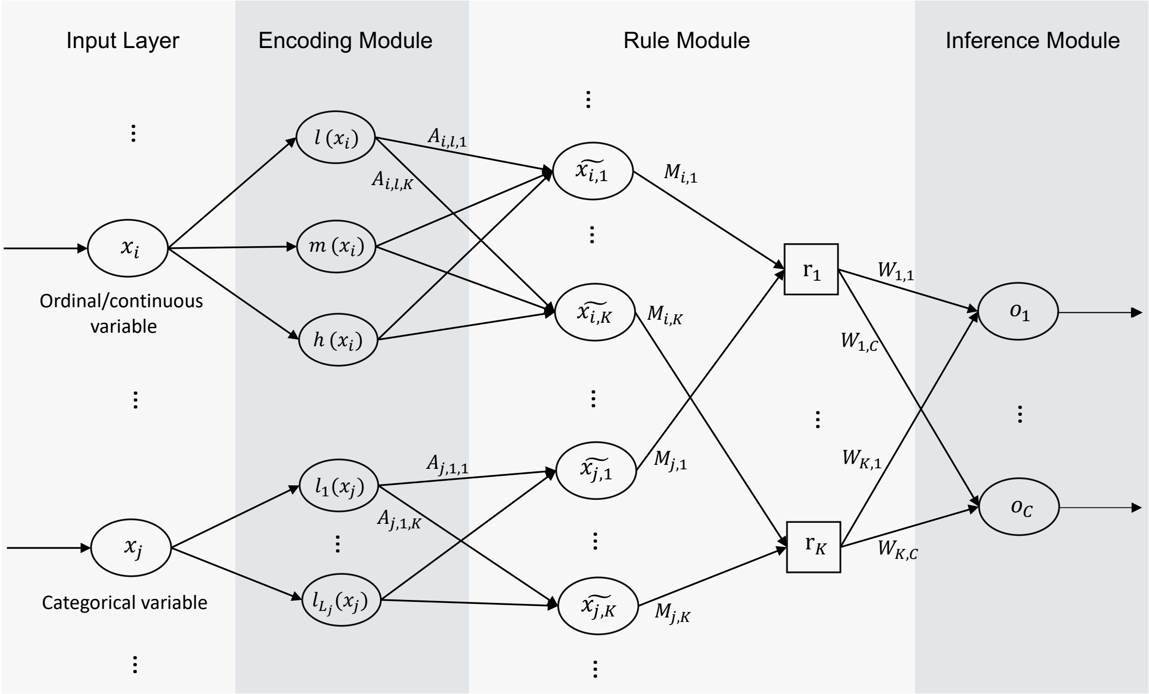


Figure 4.1: An overview of the proposed network. The nomenclatures we used in the network will be explained in §4.4.

the encoding module, an input variable is encoded into humanly understandable fuzzy concepts. In the rule module, with the trainable attention matrix and connection matrix, a limited number of fuzzy subspaces (i.e., rules) are constructed as combinations of fuzzy concepts from the encoding module. Finally, with the inference matrix and the firing strength of each rule node, the probabilities of one sample belonging to each class are calculated in the inference module. In this network, parameters in input encoding functions, subspace construction, and output inference are all trainable by gradient-based back-propagation.

Unlike prior work on fuzzy inference systems, we parametrized the membership functions and aggregation operators using ϵ, a factor that controls their smoothness. Previously, min / product and max / addition were used as T-norm and T-conorm (also called aggregation operations), respectively, though it remains unknown which is better [123, 136, 143]. Similar to the encoding functions, triangular, trapezoids, and Gaussian membership functions are all commonly used, but it is not very unclear which one is the best in fuzzy set encoding. The use of Gaussian membership functions, product, and addition enables the application of the back-propagation method for optimization. However, it is unknown whether the lack of piece-wise linearity will limit the capability of the fuzzy inference system. In addition, while the selection of membership function shape may be application-specific, several prior studies have shown that the triangular membership function is superior to other membership functions [144, 145, 146]. Previous studies also demonstrated that some practical problems are easier to solve in tropical geometry due to the piece-wise linear nature of the tropical objects [127]. As such, parametrizing the membership functions, T-norm, and T-conorm allows the model to discover optimal encoding functions and operations during the training process. Throughout the course of the optimization process, these parametrized functions are gradually updated to be closer to piece-wise linear functions, which both ensure the stability and convergence of gradient descent and results in an interpretable and accurate model. After model training, the attention matrix, connection matrix, and inference matrix can be used to interpret the model in the form of rules.

As the proposed network mimics human logic, not only can knowledge be extracted from the trained model but also existing knowledge can be integrated/transferred into the model. In this study, experiments were performed to investigate whether initializing the network with existing domain knowledge can facilitate model training.

4.4.2 Encoding module

The input variables can be either ordinal, continuous, or categorical. For ordinal and continuous variables, fuzzy theory will be used to encode variables into multiple fuzzy sets. Unlike with crisp sets, for which membership is binary, for fuzzy sets a membership value in $[0, 1]$ will be assigned to a variable's observed value for a given fuzzy set, indicating the confidence of that value belonging to the set. Fuzzy set membership approximates the fuzzy concept used by human experts during decision-making. For example, given the heart rate of a patient, the clinician may describe it as a "low" / "medium" / "high" heart rate. "Low", "medium", and "high" are the fuzzy concepts used in clinical problems. In this study, we encoded clinical ordinal/continuous variables into these three concepts. With an ordinal/continuous variable x, the membership functions $l(x), m(x), h(x)$ for "low", "medium", and "high" concepts are defined as

$$
\begin{aligned}
f_{\epsilon_1}(x) =& \epsilon_1 \log(1+\exp(x/\epsilon_1)), && \text{(4.1a)}\\
l(x) =& f_{\epsilon_1}\left(\frac{a_{i,2}-x}{a_{i,2}-a_{i,1}}\right) - f_{\epsilon_1}\left(\frac{a_{i,1}-x}{a_{i,2}-a_{i,1}}\right), && \text{(4.1b)}\\
m(x) =& f_{\epsilon_1}\left(\frac{x-a_{i,1}}{a_{i,2}-a_{i,1}}\right) - f_{\epsilon_1}\left(\frac{x-a_{i,2}}{a_{i,2}-a_{i,1}}\right) - \\
& f_{\epsilon_1}\left(\frac{a_{i,3}-x}{a_{i,4}-a_{i,3}}\right) + f_{\epsilon_1}\left(\frac{a_{i,4}-x}{a_{i,4}-a_{i,3}}\right) - 1, && \text{(4.1c)}\\
h(x) =& f_{\epsilon_1}\left(\frac{x-a_{i,3}}{a_{i,4}-a_{i,3}}\right) - f_{\epsilon_1}\left(\frac{x-a_{i,4}}{a_{i,4}-a_{i,3}}\right), && \text{(4.1d)}
\end{aligned}
$$

where $a_{i,1} < a_{i,2} < a_{i,3} < a_{i,4}$ and are trainable. With $0 < \epsilon_1 < 1$, the membership functions are differentiable, with their smoothness modulated by ϵ_1. As $\lim_{\epsilon_1 \to 0} f_{\epsilon_1}(x) = \max(0, x)$, when ϵ_1 approaches 0, the membership functions in Equation 4.1 are close to trapezoidal membership functions or triangular membership functions (if $a_{i,2}$ is close to $a_{i,3}$).

Using the defined membership functions, x_i will be encoded as membership values in three fuzzy concepts: $l(x_i), m(x_i), h(x_i)$. In this study, we used three concepts - "low", "medium", and "high" - as they are commonly used in healthcare applications. The above formulations can be easily extended to a higher number of concepts.

Categorical variables are represented via a one-hot encoding directly and no fuzzy concepts are used. We denote L_j as the number of levels of a categorical variable x_j. In this study, x_j is encoded into $l_1(x_j), l_2(x_j), \ldots, l_{L_j}(x_j)$, where only one of them has a value of 1 and all others are 0.

4.4.3 Rule module

The rule module consists of two layers in the proposed architecture. In this module, the firing strength of a number of rules (fuzzy subspaces) are calculated for the classification task and denoted as $r_1, \ldots, r_K$ in Figure 4.1, where K is the total number of rules.

4.4.3.1 The first layer

The first layer of the rule module selects the most relevant concept from each variable with respect to each rule using an attention matrix $\mathbf{A}$. $\mathbf{A}$ is the partitioned matrix formed by concatenating submatrices $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_H$, where $\mathbf{A}_h$ is the attention submatrix for the input variable x_h and $H = I + J$ is the total number of input variables, with I and J the total number of ordinal/continuous and categorical variables, respectively. For an ordinal/continuous variable x_i, the submatrix $\mathbf{A}_i$ with entries $A_{i,m,n}$ has dimension $3 \times K$, where 3 is the number of concepts for ordinal/continuous variables used in this study and K is the number of rules utilized in the network. For a categorical variable x_j, the submatrix $\mathbf{A}_j$ with entries $A_{j,m,n}$ has dimension $L_j \times K$. Thus, the attention matrix A has dimension $(3I + \sum_j L_j) \times K$.

For an ordinal/continuous variable x_i, the entry $A_{i,1,k}$ in the attention matrix represents the contribution of x_i being "low" to rule k (and similarly, $A_{i,2,k}$ for x_i being "medium" and $A_{i,3,k}$ for x_i being "high"). Entries in the attention matrix are all trainable and constrained to $[0, 1]$ by the hyperbolic tangent activation function. A higher value in $\mathbf{A}$ indicates a higher contribution. As shown in Figure 4.1, for an input variable x_i, the corresponding output from the first layer of the rule module is $\widetilde{x}_i$, a vector of length K. $\widetilde{x}_{i,k}$, the k^{th} element of $\widetilde{x}_i$, is the firing strength of x_i involved in k^{th} rule.

For an ordinal/continuous variable x_i and categorical variable x_j, $\widetilde{x}_{i,k}$, and $\widetilde{x}_{j,k}$ are calculated as

$$\widetilde{x}_{i,k} = A_{i,1,k}l(x_i) + A_{i,2,k}m(x_i) + A_{i,3,k}h(x_i), \tag{4.2a}$$

$$\widetilde{x}_{j,k} = \sum_{d=1}^{L_j} A_{j,d,k}l_d(x_j) \tag{4.2b}$$

respectively.

4.4.3.2 The second layer

The second layer of the rule module calculates rule firing strength by a connection matrix $\mathbf{M}$ of dimension $H \times K$. The k^{th} rule is constructed as a combination of $\widetilde{x}_{1,k}, \ldots, \widetilde{x}_{H,k}$ from the previous layer. An entry $M_{i,k}$ in the connection matrix $\mathbf{M}$ denotes the contribution of x_i to the k^{th} rule. Entries in the connection matrix are all trainable and constrained to $[0, 1]$ the hyperbolic tangent activation function, and a higher value indicates a higher contribution. In this layer, we define a parametrized T-norm to calculate r_k, the firing strength of the k^{th} rule.

With $0 < \epsilon_2 < 1$, let $g_{\epsilon_2} : [0, \infty) \to [0, \infty)$ and its inverse function $g_{\epsilon_2}^{-1}$ be defined as

$$g_{\epsilon_2}(x) = \frac{\epsilon_2}{1-\epsilon_2}\left(1 - x^{\frac{\epsilon_2 - 1}{\epsilon_2}}\right), \tag{4.3a}$$

$$g_{\epsilon_2}^{-1}(z) = \left(1 - \frac{1-\epsilon_2}{\epsilon_2}z\right)^{\frac{\epsilon_2}{\epsilon_2 - 1}}. \tag{4.3b}$$

The parametrized T-norm on two inputs is defined as

$$\begin{aligned} T_{\epsilon_2}\left(x,y\right) =& g_{\epsilon_2}^{-1}(g_{\epsilon_2}\left(x\right)+g_{\epsilon_2}\left(y\right)) \\ =& \left(x^{\frac{\epsilon_2-1}{\epsilon_2}}+y^{\frac{\epsilon_2-1}{\epsilon_2}}-1\right)^{\frac{\epsilon_2}{\epsilon_2-1}}, \end{aligned} \tag{4.4}$$

which has the following asymptotic behavior:

$$\lim_{\epsilon_2\to 1} T_{\epsilon_2}\left(x,y\right) = xy, \tag{4.5a}$$

$$\lim_{\epsilon_2\to 0} T_{\epsilon_2}\left(x,y\right) = \min(x,y), \tag{4.5b}$$

which means that the defined T-norm can be modulated between product and min by ϵ_2.

Using this definition of the T-norm, r_k is calculated by applying the T-norm to multiple inputs:

$$\begin{aligned} r_k =& T_{\epsilon_2}\left(\widetilde{x}_{1,k}^{M_{1,k}}, \widetilde{x}_{2,k}^{M_{2,k}}, \ldots, \widetilde{x}_{H,k}^{M_{H,k}}\right) \\ =& g_{\epsilon_2}^{-1}\left(\sum_{i=1}^{H} g_{\epsilon_2}(\widetilde{x}_{i,k}^{M_{i,k}})\right) \\ =& \left(\sum_{i=1}^{H} \widetilde{x}_{i,k}^{M_{i,k}\cdot\frac{\epsilon_2-1}{\epsilon_2}} - H + 1\right)^{\frac{\epsilon_2}{\epsilon_2-1}}. \end{aligned} \tag{4.6}$$

In Equation (4.6), entries in the connection matrix $\mathbf{M}$ are used as exponents. Taking the example of $\widetilde{x}_{1,k}^{M_{1,k}}$, a lower $M_{1,k}$ (closer to 0) means $\widetilde{x}_{1,k}^{M_{1,k}}$ is closer to 1, consequently it contributes less to r_k with the proposed T-norm. Thus, a lower value in $\mathbf{M}$ indicates a lower contribution to the rule firing strength, and vice versa.

4.4.4 Inference module

Let C denote the number of classes in the classification task. The inference layer has C nodes, one for each class, that are fully connected to the rule layer nodes. The firing strength of each node o_c is calculated using the rule firing strengths with an inference matrix $\mathbf{W}$ of dimension $K \times C$. An entry $W_{j,c}$ denotes the contribution of the k^{th} rule to the c^{th} class. Entries in the inference matrix are all trainable and positive. A higher value indicates a higher contribution. In this layer, we define a parametrized T-conorm to calculate o_c.

The parametrized T-conorm on two inputs is written as

$$Q_{\epsilon_3}(x, y) = \left(x^{\frac{1}{\epsilon_3}} + y^{\frac{1}{\epsilon_3}}\right)^{\epsilon_3}, \tag{4.7}$$

where $0 < \epsilon_3 < 1$. This T-conorm has the following asymptotic behavior:

$$\lim_{\epsilon_3 \to 1} Q_{\epsilon_3}(x, y) = x + y, \tag{4.8a}$$

$$\lim_{\epsilon_3 \to 0} Q_{\epsilon_3}(x, y) = \max(x, y), \tag{4.8b}$$

which means that the defined T-conorm can be modulated between addition and max by ϵ_3.

Using this definition of the T-conorm, o_c is calculated by applying the T-conorm to multiple inputs:

$$\begin{aligned} o_c = & Q_{\epsilon_3}(W_{1,c}r_1, W_{2,c}r_2, \ldots, W_{K,c}r_K) \\ = & \left(\sum_{k=1}^{K}(W_{k,c}r_k)^{\frac{1}{\epsilon_3}}\right)^{\epsilon_3}. \end{aligned} \tag{4.9}$$

After the calculation of $o_1, o_2, \ldots, o_C$, a softmax activation function is applied to generate probabilities $p_1, p_2, \ldots, p_C$ of being in each class, which are all in $[0, 1]$ with $\sum_{c=1}^{C} p_c = 1$.

As $\sum_{c=1}^{C} p_c = 1$, we can set the number of "valid" nodes in the inference module to $C - 1$ to avoid ambiguity in rule representation. For example, when performing binary classification $W_{:,0}$ can be set to 0 so that the model will only learn subspaces related to the positive class.

4.4.5 Network Interpretation

The proposed network can both extract rules and inject rules in a way that humans can understand. The entries in the attention matrix $\mathbf{A}$ and connection matrix $\mathbf{M}$ represent the contribution of individual concepts and individual variables to each rule. The entries in the inference matrix $\mathbf{W}$ gives the contribution of individual rules to each class.

With $\mathbf{A}$ and $\mathbf{M}$, a contribution matrix $\mathbf{S}$ can be constructed that expresses the contribution of individual concepts to each rule in the model. The matrix S is of the same dimension as attention matrix $\mathbf{A}$, i.e., it is a partition matrix formed by concatenating submatrices $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_H$. For an ordinal/continuous variable x_i, the corresponding submatrix $\mathbf{S}_i$ has dimension $3 \times K$ and for a categorical variable x_j, $\mathbf{S}_j$ has dimension $L_j \times K$. The entries $S_{i,d,k}$ of $\mathbf{S}_i$ and $S_{j,d,k}$ of $\mathbf{S}_j$ are calculated as

$$S_{i,d,k} = A_{i,d,k} \times M_{i,k}, \quad d \in \{1,2,3\}, \tag{4.10a}$$

$$S_{j,d,k} = A_{j,d,k} \times M_{j,k}, \quad d \in \{1,\ldots,L_j\}, \tag{4.10b}$$

respectively, where $k \in \{1, \ldots, K\}$.

The entry $S_{i,d,k}$ is the contribution of the d^{th} concept of x_i to the k^{th} rule. $\mathbf{S}_{:,:,k}$ encodes the construction of the k^{th} rule, while $W_{k,:}$ captures the relationship between classes and the k^{th}.

The following is a toy example further demonstrating how humanly understandable rules are represented in the network.

Given a dataset with four continuous input variable x_1, x_2, x_3, x_4 and a binary response (negative/positive), $\mathbf{A}, \mathbf{M}, \mathbf{W}$ are trained and $\mathbf{S}$ can be calculated. Let us assume that in the contribution matrix $\mathbf{S}$, $S_{1,1,1}, S_{2,3,1}, S_{2,2,2}$, and $S_{3,1,2}$ are close to 1, with all other entries close to 0. In the inference matrix $\mathbf{W}$, $W_{1,2}$ and $W_{2,2}$ are close to 1 while $W_{1,1}$ and $W_{2,1}$ are close to 0. From the given $\mathbf{S}$ and $\mathbf{W}$, we can summarize two rules from the trained network as follows:

- **IF** x_1 is low and x_2 is high, **THEN** the sample is positive;
- **IF** x_2 is medium and x_3 is low, **THEN** the sample is positive.

The above two rules are represented in $(\mathbf{S}_{:,:,1}, W_{1,:})$ and $(\mathbf{S}_{:,:,2}, W_{2,:})$, respectively. The definitions of "low", "medium" and "high" concepts can be extracted from the parameters in the encoding module. The extracted rules mimic human logic. They can be used to justify the network's decisions and contribute to knowledge discovery.

In practice, the trained model may have some redundant rules. The correlation between each pair of rules are calculated and then uses thresholds to prune redundant rules and less significant concepts.

4.4.6 Model training and network initialization

The proposed network is trained by back-propagation with an Adam optimizer. A regular cross-entropy loss $loss_{cs}$ is calculated to train the classification model. Additionally, an ℓ_1 norm-based regularization term $loss_{\ell_1}$ is added to the loss function to favor rules with a smaller number of concepts, which are more feasible to use in practice. In addition, the correlation among encoded rules is calculated as a loss term $loss_{corr}$ to avoid extracting redundant rules. The loss function can be written as:

$$loss_{total} = loss_{ce} + \lambda_1 loss_{\ell_1} + \lambda_2 loss_{corr}, \tag{4.11a}$$

$$loss_{l1} = \left\| vec(\mathbf{A}) \right\|_1 + \left\| vec(\mathbf{M}) \right\|_1, \tag{4.11b}$$

$$loss_{corr} = \sum_{i=1}^{H-1} \sum_{j=i+1}^{H} vec(\mathbf{S}_{:,:,i}) vec(\mathbf{S}_{:,:,j}) \tag{4.11c}$$

where λ_1 and λ_2 control the magnitude of the ℓ_1 norm-based regularization term and correlation based regularization term, respectively. $vec(\cdot)$ denotes the vectorization of a matrix.

In this study, for simplicity, $\epsilon_1, \epsilon_2, \epsilon_3$ are constrained to be equal. They are initialized as 0.99 at the beginning of training and are gradually reduced with the number of training steps. The scheduling of the ϵ values can be written as

$$\epsilon = max(\epsilon_{min}, \epsilon \cdot \gamma^{training\_steps}), \tag{4.12}$$

where γ is the decay rate that can be tuned as a hyperparameter. From our preliminary analysis, $\gamma = 0.999$ usually is a good choice. ϵ_{min} is another hyperparameter, whose optimal value varies with different applications. The hyperparameter tuning strategy will be discussed in the next section. Our experiments show that starting with $\epsilon = 0.99$ and reducing ϵ improves model optimization (as discussed in §4.5.1).

Before model training, trainable parameters will be randomly initialized. To improve performance, especially when the size of the training dataset is small, practical rules from domain knowledge can be used to initialize the network. Revisiting the toy example in §4.4.5, if the extracted rules were instead previously known within the application domain, the matrices $\mathbf{A}, \mathbf{M}$, and $\mathbf{W}$ in the network could then be initialized as:

- $\mathbf{A}$: $A_{1,1,1}, A_{2,3,1}, A_{2,2,2}, A_{3,1,2}$ have a higher value and other entries in $A_{:,:,1}$ and $A_{:,:,2}$ have a lower value;
- $\mathbf{M}$: $M_{1,1}, M_{2,1}, M_{2,2}, M_{3,2}$ have a higher value and other entries in $M_{:,1}$ and $M_{:,2}$ have a lower value;
- $\mathbf{W}$: $W_{1,2}, W_{2,2}$ have a high value and $W_{1,1}, W_{2,1}$ have a low value;
- Other entries in $\mathbf{A}$, $\mathbf{M}$, and $\mathbf{W}$ are randomly initialized.

4.4.7 Evaluation strategy

For synthetic datasets, a 10-fold cross-validation was used to evaluate model performance; and for heart failure dataset, the proposed data split in Table 4.2 was randomly repeated for 10 times to evaluate the model. A random search algorithm was applied using the training set and validation set for hyperparameter tuning, including learning rate, batch size, λ_1, λ_2, and ϵ_{min}. The model trained with the optimal combinations of hyperparameters was then evaluated on the test set. The performance of the proposed network will be presented as the average and standard deviation (std) from 10 iterations.

For comparison, several popular "black box" ML algorithms were chosen, including random forest, support vector machine (SVM), and XGBoost. In addition, several interpretable models were chosen including logistic regression, decision tree, and explainable boosting machine (EBM, a type of generalized addictive models) [147], and a fuzzy inference classifier [148]. Similarly, a 10-fold cross-validation was used to evaluate the performance of those models. Hyperparameters were also tuned using a training set and validation set. Detailed implementation information for these models is described in Appendix G.

Accuracy, recall, precision, F1, AUC and AUPRC were calculated to evaluate the performance of the trained classifiers.

4.5 Results and Discussion

4.5.1 Synthetic dataset 1 ($N = 400$)

Let N denote the number of observations in a given dataset. Several experiments were performed with differently sized simulated datasets. In this section, we discuss the performance of the proposed method on synthetic dataset 1 when $N = 400$.

The first experiment starts with $N = 400$.The proposed network was trained using

Model	Accuracy	Recall	Precision	F1	AUC
$\epsilon_{min} = 0.8$	0.955 (0.025)	0.911 (0.073)	0.955 (0.038)	0.883 (0.040)	0.986 (0.016)
$\epsilon_{min} = 0.4$	0.959 (0.030)	0.904 (0.073)	**0.972 (0.035)**	0.888 (0.048)	0.991 (0.010)
$\epsilon_{min} = 0.2$	0.961 (0.026)	**0.919 (0.087)**	0.968 (0.039)	**0.892 (0.045)**	**0.992 (0.008)**
$\epsilon_{min} = 0.1$	0.901 (0.053)	0.856 (0.146)	0.865 (0.089)	0.803 (0.093)	0.949 (0.056)
Fixed $\epsilon = 0.8$	**0.966 (0.023)**	0.903 (0.083)	0.964 (0.019)	0.886 (0.037)	0.978 (0.019)
Fixed $\epsilon = 0.4$	0.939 (0.040)	0.867 (0.086)	0.948 (0.056)	0.857 (0.064)	0.964 (0.024)
Fixed $\epsilon = 0.2$	0.786 (0.041)	0.519 (0.190)	0.803 (0.109)	0.558 (0.132)	0.819 (0.117)
Fixed $\epsilon = 0.1$	0.789 (0.062)	0.552 (0.237)	0.689 (0.255)	0.560 (0.216)	0.855 (0.081)

Table 4.3: Performance of the proposed model on the synthetic dataset 1 with $N = 400$ with different ϵ settings using 10-fold cross-validation. For the first half of rows, ϵ starts with 0.99 and gradually reduced to ϵ_{min} during the training. For the second half of rows, the value of ϵ didn't change during the training process.

80% of the data and tested on 20% of the data. The percentage of positive samples is 34.25%, and the percentages of samples with Rule A, Rule B, Rule C, Rule D, Rule E are 8.25%, 7.50%, 9.00%, 2.00%, and 10.75%, respectively.

Table 4.3 depicts the performance of the proposed algorithm with different ϵ_{min} on the test sets from a 10-fold cross-validation. We can observe that model training benefited from decreasing ϵ_{min} from 0.8 to 0.2, but the performance of the trained model decreased when ϵ_{min} was decreased to 0.1. We also evaluated the model with a fixed ϵ, rather than gradually decreasing it from 0.99. While fixing ϵ at 0.8 leads to comparable performance with the model using $\epsilon_{min} = 0.8$, the performance of the models with a smaller fixed ϵ value decreased significantly. Our results show the effectiveness of the algorithm that gradually decreases ϵ during the training. Using this dataset, the proposed network with a reasonable degree of piecewise linearity has a better performance.

Table 4.4 describes the performance of the proposed method where ϵ_{min} is tuned on the validation set in each iteration. The performance of the proposed network is compared with that of other machine learning algorithms. From Table 4.4, we can see that the proposed network achieved significantly better performance than other interpretable models and had comparable performance to the XGBoost model, which is the best among the other established machine learning algorithms.

Model	Accuracy	Recall	Precision	F1	AUC
Proposed	0.960 (0.023)	0.933 (0.054)	0.953 (0.060)	0.893 (0.032)	0.994 (0.005)
EBM	0.835 (0.027)	0.678 (0.060)	0.807 (0.060)	0.688 (0.045)	0.924 (0.018)
Logistic Regression	0.724 (0.029)	0.344 (0.078)	0.692 (0.098)	0.413 (0.070)	0.701 (0.065)
Naïve Bayes	0.734 (0.032)	0.363 (0.089)	0.721 (0.114)	0.434 (0.082)	0.803 (0.035)
Decision Tree	0.933 (0.046)	0.907 (0.056)	0.901 (0.090)	0.855 (0.064)	0.938 (0.040)
Fuzzy Inference	0.680 (0.036)	0.456 (0.102)	0.540 (0.076)	0.441 (0.071)	0.668 (0.056)
Random Forest	0.924 (0.015)	0.826 (0.062)	0.944 (0.037)	0.832 (0.028)	0.981 (0.006)
XGBoost	0**.977 (0.013)**	**0.959 (0.031)**	**0.975 (0.028)**	**0.919 (0.020)**	**0.996 (0.003)**
SVM	0.821 (0.038)	0.641 (0.076)	0.796 (0.077)	0.661 (0.061)	0.897 (0.026)

Table 4.4: Performance comparison on the synthetic dataset 1 with $N = 400$ using 10-fold cross-validation.

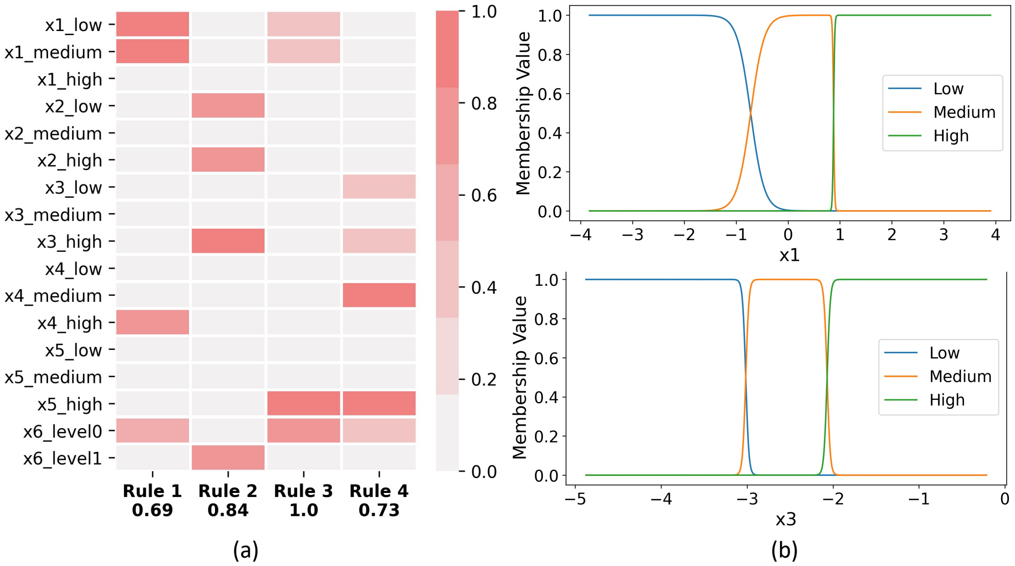


Figure 4.2: Interpretation of a trained model on synthetic dataset 1 with $N = 400$. (a) Visualization of four rules contributing to the positive class, which are summarized from the trained model. Rules are visualized in individual columns with the corresponding contribution. The concept names are given as row names. For example, "x1_low" means "the value of $x1$ is low". The contribution of individual concepts to individual rules are shown in color; (b) Membership functions for "low", "medium", and "high" concepts of x_1, x_3 in the encoding module, respectively.

To examine the proposed network's ability to learn rules from the dataset, we summarized rules contributing to the positive class from a trained network. Those rules are visualized in Figure 4.2 (a). Comparing the learned rules with rules in Section 4.3.1.1, we can observe that Rule 1 corresponds to Rule C; Rule 2 corresponds to a union of Rule A and Rule B; Rule 3 corresponds to Rule E; and Rule 4 is closest to Rule D. Membership functions of the variables involved in Rule 1 and Rule 2 are visualized in Figure 4.2 (b) and we can observe a great match. For example, the membership value of x_2 to the "low" concept is high when x_2 smaller than 3.7 and the membership value of x_2 to the "high" concept is high when x_2 is larger than 6.2. Simple thresholds were used to construct synthetic dataset 1, and for this reason the fuzzy regions in the membership functions are very narrow. From the interpretation in Figure 4.2, the trained model learned the majority of rules used to construct the dataset. Rule 4 is close to Rule D but with two additional concepts that are misidentified as related to the class. This may be due to only 2.00% of samples in the dataset being consistent with Rule D, making it more challenging to learn from the data. In addition, from Figure 4.2 (a), concepts from x_7 and x_8 are not shown because their significance to learned rules is too low. This demonstrates that the proposed network can identify and exclude irrelevant variables.

4.5.2 Synthetic dataset 1 ($N = 50$)

In the second experiment, we used synthetic dataset 1 with $N = 50$. The percentage of positive samples is 42.00%, and the percentages of samples with Rules A-E are 14.00%, 14.00%, 4.00%, 4.00%, and 12.00%, respectively. In this experiment, we investigated the performance of the proposed network with a small training set and if initiating the network with existing knowledge would enable the model to learn more accurate rules.

Table 4.5 has three blocks, presenting the performance of the proposed networks, established interpretable ML methods, and established black-box ML methods on synthetic dataset 1 ($N = 50$), respectively. The first block shows the performance of the proposed network without and with existing knowledge. The performance of the proposed network with random initialization is shown in the first row of the first block, followed by the performance of the proposed network initialized with existing knowledge (rules). Rules A through E are fully correct as described in Section 4.3.1.1 while Rules F through H are partially correct. In practical applications, it is very rare that the ground truth rule is available. As such, in this experiment, we only initialized $\mathbf{A}$, $\mathbf{M}$, and $\mathbf{W}$, while the parameters in the membership functions were randomly initialized. In addition, to investigate whether inexact domain knowledge can facilitate model training, we proposed the following three rules and assumed they lead to a positive class:

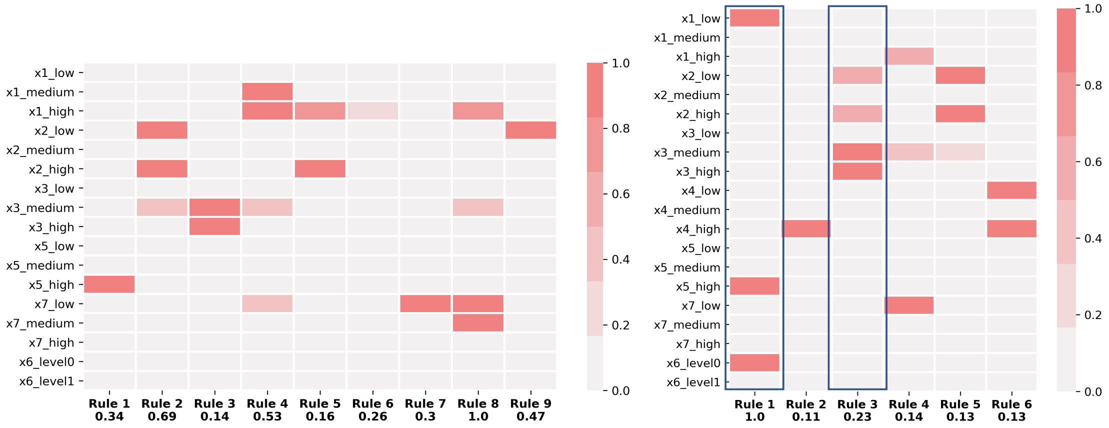
- Rule F: x_2 is "low" and $x_6 = 1$;
- Rule G: x_1 is "low" and x_5 is "low" and $x_6 = 0$;
- Rule H: x_1 is "low" and x_5 is "high" and $x_6 = 0$ and x_7 is "high";

Rule F, G, and H are only partially correct. Compared with ground truth Rule A, the "high" concept of x_3 is missing in Rule F. In Rule G, x_5 should be "high" rather than "low" as in Rule E. In Rule H, "high" concept of x_7 is actually irrelevant to the class.

From Table 4.5, we first observe that because of the reduction in the size of the training set, performance decreased. Still, XGBoost achieves the best performance, and the proposed network with random initialization has a comparable performance to XGBoost. Second, we observe that the improvement can be achieved when the network was initialized with Rules A through E. Third, the model's performance increased when it was initialized with partially correct rules. This indicates that

Model	Accuracy	Recall	Precision	F1	AUC
Proposed (None)	0.640 (0.143)	0.550 (0.292)	0.518 (0.249)	0.473 (0.236)	0.688 (0.213)
Proposed (Rule A)	0.670 (0.110)	0.575 (0.275)	0.543 (0.238)	0.504 (0.223)	0.710 (0.188)
Proposed (Rule B)	0.670 (0.135)	0.600 (0.255)	0.646 (0.211)	0.535 (0.170)	0.658 (0.183)
Proposed (Rule C)	0.690 (0.104)	0.625 (0.202)	0.658 (0.197)	0.566 (0.129)	0.698 (0.158)
Proposed (Rule D)	0.730 (0.142)	**0.675 (0.251)**	0.658 (0.282)	**0.607 (0.225)**	0.710 (0.194)
Proposed (Rule E)	0.700 (0.190)	0.600 (0.229)	0.710 (0.259)	0.573 (0.202)	**0.740 (0.191)**
Proposed (Rule F, partially correct)	0.680 (0.183)	0.600 (0.200)	0.665 (0.278)	0.565 (0.196)	0.688 (0.206)
Proposed (Rule G, partially correct)	0.700 (0.210)	0.625 (0.280)	0.605 (0.308)	0.566 (0.276)	0.652 (0.213)
Proposed (Rule H, partially correct)	**0.750 (0.112)**	0.575 (0.195)	**0.775 (0.197)**	0.593 (0.176)	**0.740 (0.152)**
EBM	0.650 (0.120)	0.500 (0.224)	0.562 (0.260)	0.469 (0.192)	0.670 (0.151)
Logistic Regression	0.610 (0.145)	0.425 (0.275)	0.512 (0.339)	0.395 (0.236)	0.583 (0.181)
Naïve Bayes	0.640 (0.120)	0.475 (0.208)	0.552 (0.159)	0.457 (0.178)	0.629 (0.174)
Decision Tree	0.530 (0.200)	0.425 (0.317)	0.398 (0.263)	0.361 (0.261)	0.527 (0.203)
Fuzzy Inference Classifier	0.520 (0.117)	0.525 (0.208)	0.416 (0.120)	0.413 (0.146)	0.550 (0.103)
Random Forest	0.650 (0.081)	0.475 (0.236)	0.580 (0.275)	0.450 (0.176)	0.619 (0.168)
XGBoost	0.650 (0.186)	0.600 (0.300)	0.591 (0.275)	0.521 (0.238)	0.675 (0.187)
SVM	0.580 (0.075)	0.125 (0.230)	0.250 (0.403)	0.130 (0.204)	0.521 (0.173)

Table 4.5: Performance comparison of the proposed network and other established ML methods on the synthetic dataset 1 with $N = 50$ using 10-fold cross-validation.



(a)

(b)

Figure 4.3: Rules contributing to the positive class learned by the trained proposed network on the synthetic dataset 1 with $N = 50$. (a) Model's parameters were randomly initialized; (b) $A_{:,:,1}$, $M_{:,1}$, $W_{1,:}$ were initialized by Rule H while other entries were initialized with the same values in (a).

existing domain knowledge can help with model training even when the rules are vague and/or inexact.

In Figure 4.3, we interpret and visualize the model trained from scratch and the model initialized with Rule H. From Figure 4.3 (a), we find that the learned rules are less accurate compared with Figure 4.2 (a) because of the reduced size of the training set. In Figure 4.3 (b), Rule 1 shows that even though the model was initialized with a partially correct rule, the model can identify that “high” x_7 doesn’t contribute to the classification; and Rule 3 indicates that initializing the model with existing knowledge can also facilitate the model learning other rules.

4.5.3 Synthetic dataset 2 ($N = 400$)

The responses in synthetic dataset 1 were constructed by rules, where a rule-based or tree-based ML algorithm may be more favorable. Therefore, responses in synthetic dataset 2 were built from a non-linear function to further explore the capacity of the proposed network in function approximation. The performance comparison of different ML models is presented in Table 4.6. From the table, we can see that SVM achieved the best average AUC. The performance of the proposed network is comparable with the random forest model, XGBoost, and EBM, holding the second place. But considering that the standard deviation of the SVM model is relatively high, SVM model doesn’t have a significantly better performance.

Rules extracted from the trained proposed network are presented in Figure 4.4. We see that these rules capture meaningful information. Observations in this dataset were annotated as positive if $(x_1 + 0.5x_2 + x_3)^2/(1 + e^{x_6} + 2x_7) < 1$. Rule 1 shows that “high” levels of x_6 and x_7 lead to the positive class. In this dataset, x_1, x_2, and x_3 were simulated as: $x_1 \sim \mathcal{N}(0, 2)$, $x_2 \sim \mathcal{N}(5, 3)$, and $x_3 \sim \mathcal{N}(-1, 5)$. As such, a “high” x_1 and ”low” x_3 can lead $(x_1 + 0.5x_2 + x_3)^2$ to a small value. A ”low” or ”medium” x_1 and ”medium” x_3 is another combination that can lead $(x_1+0.5x_2+x_3)^2$

Model	Accuracy	Recall	Precision	F1	AUC
Proposed	0.735 (0.044)	0.708 (0.050)	0.719 (0.060)	0.666 (0.044)	0.821 (0.037)
EBM	0.736 (0.028)	0.686 (0.047)	0.731 (0.044)	0.660 (0.028)	0.826 (0.042)
Logistic Regression	0.746 (0.046)	0.703 (0.084)	0.738 (0.053)	0.671 (0.058)	0.806 (0.049)
Naïve Bayes	0.723 (0.047)	0.665 (0.078)	0.720 (0.068)	0.642 (0.054)	0.807 (0.044)
Decision Tree	0.674 (0.046)	0.616 (0.069)	0.660 (0.058)	0.589 (0.052)	0.679 (0.050)
Fuzzy Inference	0.654 (0.048)	0.408 (0.090)	0.721 (0.076)	0.475 (0.084)	0.761 (0.037)
Random Forest	0.734 (0.040)	0.692 (0.030)	0.726 (0.058)	0.660 (0.034)	0.827 (0.035)
XGBoost	0.734 (0.043)	0.705 (0.072)	0.714 (0.043)	0.662 (0.054)	0.837 (0.033)
SVM	**0.781 (0.074)**	**0.741 (0.077)**	**0.780 (0.094)**	**0.712 (0.079)**	**0.871 (0.066)**

Table 4.6: Performance comparsion on the synthetic dataset 2 with $N = 400$.

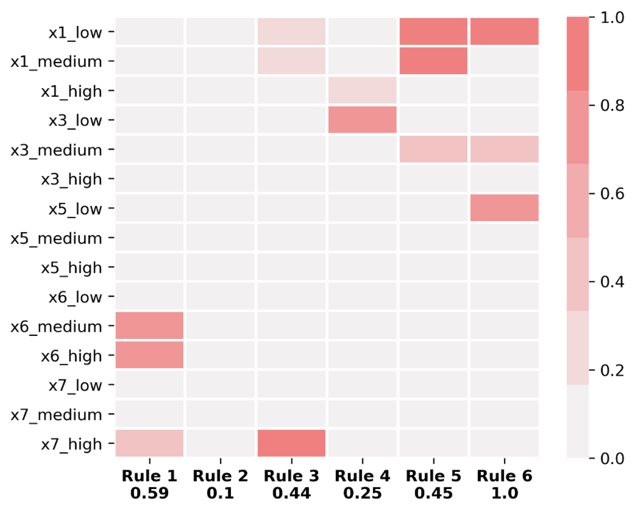


Figure 4.4: Interpretation of a trained model on the synthetic dataset 2 with $N = 400$.

to a small value. As expected, Rules 4 and 5 unite concepts from x_1 and x_3. From this analysis, we observe that the proposed network can learn simple rules in a format that humans can understand from a dataset that was constructed with a complicated non-linear function.

4.5.4 HF dataset

We applied the proposed network to identify patients that are eligible for advanced therapies. From Table 4.7, initializing the network with existing knowledge can greatly facilitate model performance. The proposed method had a lower AUC compared with EBM, Random Forest, and XGBoost. However, those models have

Model	Accuracy	Recall	Precision	F1	AUC
Proposed (None)	0.735 (0.047)	0.500 (0.069)	0.384 (0.059)	0.386 (0.047)	0.730 (0.042)
Proposed (with existing rules)	0.718 (0.035)	**0.645 (0.125)**	0.410 (0.045)	**0.452 (0.043)**	0.753 (0.025)
EBM	0.787 (0.018)	0.122 (0.032)	0.557 (0.150)	0.173 (0.049)	0.795 (0.034)
Logistic Regression	0.783 (0.011)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.541 (0.062)
Naïve Bayes	0.781 (0.012)	0.012 (0.013)	0.383 (0.435)	0.019 (0.020)	0.496 (0.025)
Decision Tree	0.787 (0.013)	0.072 (0.043)	0.600 (0.221)	0.108 (0.061)	0.593 (0.047)
Fuzzy Inference Classifier	0.669 (0.182)	0.422 (0.379)	0.454 (0.170)	0.262 (0.130)	0.739 (0.048)
Random Forest	0.782 (0.011)	0.004 (0.012)	0.029 (0.086)	0.005 (0.016)	**0.834 (0.016)**
XGBoost	**0.792 (0.013)**	0.079 (0.035)	**0.659 (0.104)**	0.123 (0.051)	0.792 (0.029)
SVM	0.746 (0.037)	0.116 (0.068)	0.291 (0.181)	0.130 (0.079)	0.636 (0.069)

Table 4.7: Performance comparison of the proposed network and other established ML methods on the HF dataset using 10-fold cross-validation.

low values in recall and F1-score, which means they tend to classify all samples as "negative". In addition, those three methods achieved very high values on the validation set for all metrics, and this indicates severe overfitting on the validation set. Figure 4.5 shows the generalization error between validation set and test set for five ML models. We can find the generalization errors for EBM, Random Forest, and XGBoost are very high. In contrast, the proposed method had a significantly smaller generalization error.

This finding suggests that AUC should not be the only metric used for model evaluation. AUC is good at summarising the differential ability of ML models without a fixed classification threshold. However, in practice, a threshold is usually needed for a classifier to provide recommendations in diagnosis, treatment, or outcome prediction. During the training process, an optimal threshold will be determined on the training and validation set. In this study, the gap between the recall and F1-score on the validation set and test set indicates the generalizability of the tuned threshold. From our result, EBM, Random Forest, and XGBoost all overfitted the validation set significantly. It also suggests that the model should be only tuned on the validation set, and nested cross-validation is necessary for model evaluation.

From Figure 4.5, the proposed method achieved a significantly smaller generalization error. It is an important observation as the model generalizability is also a

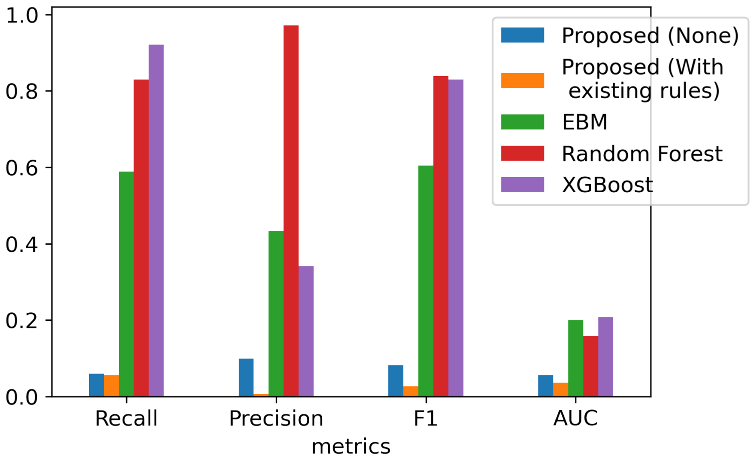


Figure 4.5: Generalization error between the validation set and test set.

primary concern of applying AI techniques in clinical applications. Notably, integrating existing domain knowledge can not only improve the classification performance, but also further reduce the generalization error. It is an important finding, showing the value of existing knowledge. While the collected existing knowledge are approximate and may not be 100% accurate, they are from experienced clinical experts and has a better generalizability. The capability of the proposed algorithm in learning existing domain knowledge make it promising in many clinical and also non-clinical applications, especially when data collection is expensive.

From Figure 4.5, the proposed method achieved a significantly smaller generalization error. It is an important observation as the model generalizability is also a primary concern of applying AI techniques in clinical applications. Notably, integrating existing domain knowledge can not only improve the classification performance but also further reduce the generalization error. It is an important finding, showing the value of existing knowledge. While the collected existing knowledge are approximate and may not be 100% accurate, they are from experienced clinical experts and has better generalizability. The capability of the proposed algorithm in learning existing domain knowledge makes it promising in many clinical and non-clinical applications, especially when data collection is expensive.

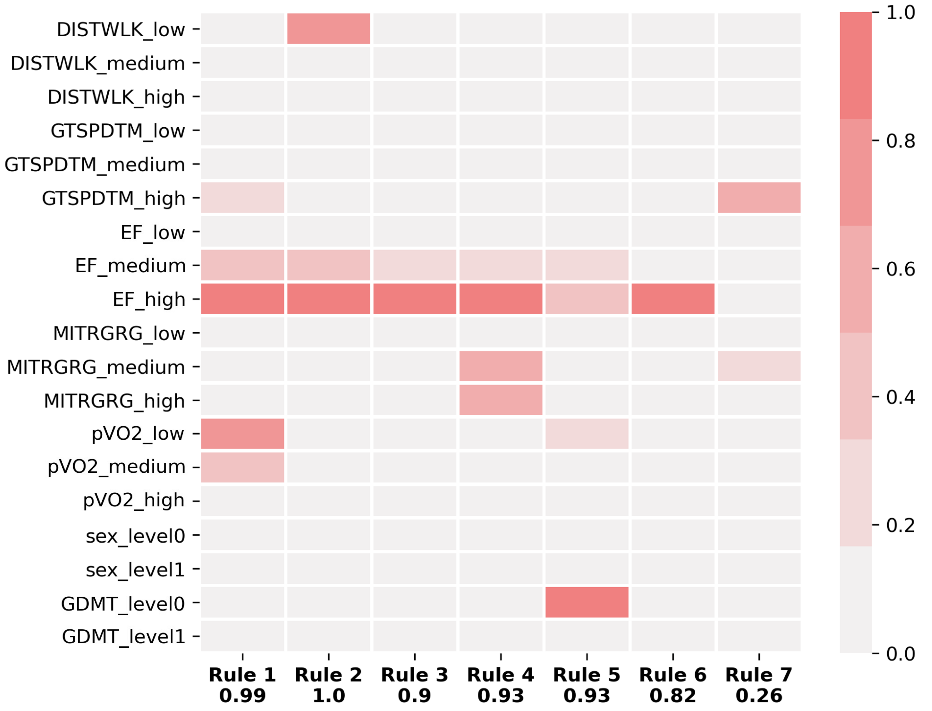


Figure 4.6: Interpretation of a trained model on the HF dataset.

4.5.5 Limitations and future work

The proposed network will be further extended and explored in future work. In the current optimization method, we use the same smoothness factor for encoding membership functions and aggregation operators. A simple linear decrease with the training steps was performed to optimize the smoothness factor. In future work, we will explore the possibility of optimizing the smoothness factors individually in respective modules with a more effective optimization method.

CHAPTER V

Additional work in solving challenges of applying machine learning and deep learning algorithms to practical problems

5.1 Filter Pruning Technique

5.1.1 Introduction

In CNN, a larger network tends to have a high capacity to find the complex functions but at the cost of having highly redundant parameters. The filters, visual interpretation of weights, in the network often have similar patterns and some of them have noise rather than distinct features. The redundancy in CNN will impair the model generalizability and accompanies unnecessary computation cost. The real-time application of DL techniques is often restricted by computation cost, memory storage and energy efficiency. The desktop system may have the luxury of burning 250W of power for neural network computation, but embedded processors targeting the automotive market must fit within a much smaller power and energy envelope. Therefore, a lightweight and computation-efficient system is important for real time applications.

Inspired by [149], we propose a Scale Module for filter pruning in an automatic manner and efficiently reduced the size of the network and inference time. The proposed method is used in an application of driver's drowsiness identification.

Drowsiness can be dangerous when people are performing tasks requiring constant concentration, such as operating high-tech machinery or motor vehicles. An active drowsiness monitoring and alert system can reduce fatigue-related incidents and save lives. In this application, we focus on using visual cues to identify drowsiness. A 3D convolutional network was developed that extracts both spatial and temporal features of consecutive frames and makes predictions on attention status. The proposed Scale Module is integrated into the 3D convolutional network to help reduce the network size and speed up the inference phase.

5.1.2 Dataset

We used the Drowsiness Detection Dataset collected by Weng *et al.* [150] from National Tsing Hua University. This dataset contains videos from 36 subjects, where they played a plain driving game with and without glasses/sunglasses. All videos are grayscale and captured by a digital camera at a resolution of 640×480 pixels and 30 FPS in the daytime while 15 FPS at night. The dataset was divided into the training set ($n = 18$), evaluation set ($n = 4$), and test set ($n = 14$) by the dataset creators. The training set and evaluation set are available for public use. Overall, the training set contains 360 videos, with an average duration of 90 seconds, while the evaluation set contains 20 videos that range in duration from 2-10 minutes. A binary annotation is provided for each video: driver status - drowsy/stillness. We used the training set to train our model and performed 6-fold cross-validation. The evaluation set is used to measure the final model's performance and to compare with other published results. In each cross-validation fold, videos from 15 subjects are used for training and videos from the remaining 3 subjects are used for validation.

5.1.3 Methods

5.1.3.1 Drowsiness detection system

Frames were first extracted from each video, after which facial regions were detected using OpenFace [151], an open-source framework that implements state-of-the-art facial behavior analysis algorithms. The face bounding box generated for each frame was extended to a square box to preserve the original ratio of the face and was resized to 64 × 64. As only subtle differences exist among frames in a short time, we sub-sampled these frames with a step size of 10 when the FPS is 30 and 5 when the FPS is 15. The sample fed into the 3D CNN is a sequence of 10 consecutive frames after sub-sampling and it abstracts information in about 3.3 seconds. The annotation of the last frame is the label. Thus, the prediction of one frame in the evaluation phase is based on the information in about 3.3 seconds before this frame. In this study, for a video of 90 seconds (FPS=30), about 260 samples were generated.

A 3D CNN was designed to extract features related to facial expression and head motion from the sequential frames. While a 2D kernel can only extract spatial features, a 3D kernel has the potential to learn spatio-temporal features.

Let $F : \mathbb{Z}^3 \to \mathbb{R}$ be a discrete function. Let $\Omega_r = [-r, r]^3 \cap \mathbb{Z}^3$ and let $k : \Omega_r \to \mathbb{R}$, be a discrete filter of size $(2r+1)^3$. The discrete 3D convolution operation $*$ can be defined as:

$$(F * k)(\mathbf{p}) = \sum_{\mathbf{s}+\mathbf{t}=\mathbf{p}} F(\mathbf{s})k(\mathbf{t}), \tag{5.1}$$

where $\mathbf{t}$ are from $[-r, r]^3 \cap \mathbb{Z}^3$ and $\mathbf{s}, \mathbf{p}$ are from $\mathbb{Z}^3$.

Our 3D CNN consists of four convolutional layers, three max-pooling layers and two fully-connected layers.

5.1.3.2 Filter pruning with Scale Module

To assist with filter pruning, we introduced a sub-network, named Scale Module, to weight filters. The design of the Scale module is inspired by the squeeze-and-excitation module proposed in [149], which is intended to model the interdependence of activation maps and perform dynamic channel-wise feature recalibration. Unlike in [149], our scale module is used to infer the importance of filters in convolutional layers and provide guidance for the following filter pruning.

As shown in Figure 5.1, the proposed Scale Module is added beside the regular convolutional layer and can be adapted to any CNN structure. The inputs of the Scale Module are the weights of the filter bank in one convolutional layer. The l_1 norm of each vectorized filter is calculated and will be passed through two consecutive fully-connected layers. After that, the scale vector is computed by an element-wise sigmoid function over the output from the last fully-connected layer.

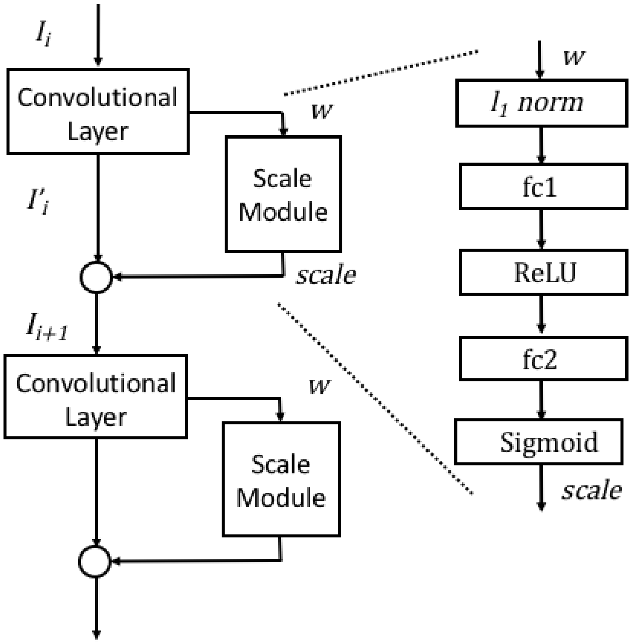


Figure 5.1: A diagram of an extended convolutional layer with the proposed Scale Module. The left side shows the overview of integrating the Scale Module into a regular convolutional layer and the right side gives the architecture of the Scale Module.

The right side of Figure 5.1 shows the architecture of the Scale Module. If the i^{th} convolutional layer has m filters then $\mathbf{W}_i = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m]$ is the weight matrix with a shape of $m \times depth \times height \times width \times channel$ for a 3D convolution operation, or $m \times height \times width \times channel$ for a 2D convolution operation. The output from the l_1 *norm* operation performed on $\mathbf{W}_i$ is a vector of length m, which can be written as $[\|vec(\mathbf{w}_1)\|_1$, $\|vec(\mathbf{w}_2)\|_1$, ..., $\|vec(\mathbf{w}_m)\|_1]$, where $vec(\cdot)$ is the vectorization operation. The fc1 layer has $\frac{m}{16}$ neurons and the fc2 layer has m neurons. After the sigmoid function, the output $\mathbf{scale}_i = [scale_1, scale_2, \ldots, scale_m]$ indicates the importance of each filter in the i^{th} convolutional layer, where all elements are mapped between zero and one. In general, the calculation of $\mathbf{scale}_i$ for the i^{th} convolutional layer can be written as:

$$\mathbf{scale}_i = S(\mathbf{q}_{2,i}^{\intercal} R(\mathbf{q}_{1,i}^{\intercal} f(\mathbf{W}_i) + \mathbf{b}_{1,i}) + \mathbf{b}_{2,i}), \tag{5.2}$$

where S is the sigmoid function and R is the ReLU function, f is the described l_1 *norm* operation, i is the index of the convolutional layer, $\mathbf{q}_{1,i}, \mathbf{b}_{1,i}, \mathbf{q}_{1,i}, \mathbf{b}_{2,i}$ are weights and biases of fc1 and fc2 in the i^{th} Scale Module, respectively.

The general format of a regular convolution operation between I_i and j^{th} filter $\mathbf{w}_j$ can be written as:

$$I'_{i+1,j} = I_i * \mathbf{w}_j, \tag{5.3}$$

After introducing the Scale Module, the output is calculated as:

$$I_{i+1,j} = scale_j I'_{i+1,j}, \tag{5.4}$$

From equations (5.3) and (5.4), the output from the extended convolutional layer can be re-written as:

$$I_{i+1,j} = I_i * scale_j \mathbf{w}_j, \tag{5.5}$$

Our design proposes to automatically assign weights for filters in the convolutional layers by using the Scale Module. From previous studies, the magnitude of filters can indicate their importance but the relationship may be too complex to be differentiated by a threshold. Using two fully connected layers, we are able to approximate the function between the magnitude and importance of filters. It will also consider the dependence among filters in the same layer. The sigmoid function acts as a 'gate' and will map the scale value to one for the most essential filters and to zero for redundant filters. The initiation value of $\mathbf{b}_{\cdot,2}$ is a vector of ones, and therefore before training, the initial scale values for all filters are approximately 0.73, i.e.,$R(1)$. In this way, all filters are regarded as non-redundant after the initialization and will be updated in the training phase. From equation (5.5), if $scale_j$ is close to zero, $I_{i+1,j}$ will also be close to zero, and the effect of $\mathbf{w}_j$ is diminished, while if $scale_j$ is close to 1, the effect of $\mathbf{w}_j$ is maintained. After the model is trained, filters with a small scale value can be removed directly with little loss in the original accuracy. As opposed to other filter pruning techniques, fine-tuning is not required after redundant filters are removed. In the experiment, we removed filters with a scale value smaller than 0.5. The histograms in Figure 4 show that scale values are either near zero or near one. Thus, the performance of the network is completely independent of any reasonable choice of threshold.

To facilitate the training process, the loss function of a CNN with J convolutional layers is extended as:

$$loss = loss_{ori} + \gamma \sum_{j=1}^{J} \parallel \mathbf{scale}_k \parallel_1, \tag{5.6}$$

where $loss_{ori}$ is the loss function of the regular CNN and $loss$ is the loss function after the Scale Module is introduced, $\mathbf{scale}_j$ denotes the scale vector in the j^{th} convolutional layer, and γ is a constant to control the power of filter pruning. In the next section,

	Accuracy (%)				Parameter	FLOP Reduc-
	Scaled Model	Scale-Pruned Model	Baseline	l_1 norm-Pruned	Reduction (%)	-tion (%)
$\gamma = 10^{-1}$	76.3 (3.0)	75.3 (2.9)	75.8 (3.7)	73.8 (2.9)	76.1 (1.2)	80.0 (1.9)
$\gamma = 10^{-2}$	76.6 (2.4)	76.3 (1.9)	75.8 (3.7)	74 (3.1)	76.2 (1.2)	76.9 (2.2)
$\gamma = 10^{-3}$	77.5 (3.0)	77.6 (2.4)	75.8 (3.7)	74.9 (2.5)	74.2 (1.8)	73.6 (1.7)
$\gamma = 10^{-4}$	77.4 (2.7)	77.4 (2.3)	75.8 (3.7)	75.2 (3.0)	54.7 (3.6)	47.2 (3.9)

Table 5.1: Evaluation of the proposed filter pruning method using 6-fold cross-validation on the training set. The parameter/FLOP reductions were calculated using the number of parameters/FLOPs after pruning divided by that before pruning.

we will compare the filter pruning performance under different values of γ.

5.1.4 Results and discussion

First, we performed a 6-fold cross-validation using the training set to evaluate our proposed filter pruning method using different γ values of $10^{-1}, 10^{-2}, 10^{-3}$, and 10^{-4}. In each fold, the 3D CNN model integrated with the Scale Module (Scaled Model) and the 3D CNN without the Scale Module (Baseline) were built. All weights were initialized according to the Xavier scheme [152] and biases were initialized with zeros except for the fc2 in the Scale Module as described in section III. The Adam optimizer was used to minimize the loss with an initial learning rate of 10^{-4}. The L2 weight decay regularization of 10^{-4} was used to improve the generalizability of the model.

After the models were trained, filters with scale values smaller than 0.5 were removed in the Scaled Models (Scale-Pruned). For comparison, the exact same number of filters in each layer of the Baseline model were removed, either randomly (Random-Pruned) or based on the l_1 norm of the filters (l_1 norm-Pruned) as described in [153]. The Random-Pruned Baseline and l_1 norm-Pruned Baseline were further fine-tuned with a learning rate of 10^{-8} as suggested by the literature while no fine-tuning was performed for the Scale-Pruned Model.

The average accuracies and reductions in the number of parameters and FLOPs after filter pruning are listed in Table 5.1. The results show that the average accuracies of Scaled Models are higher than that of the Baseline with less than 1% increase

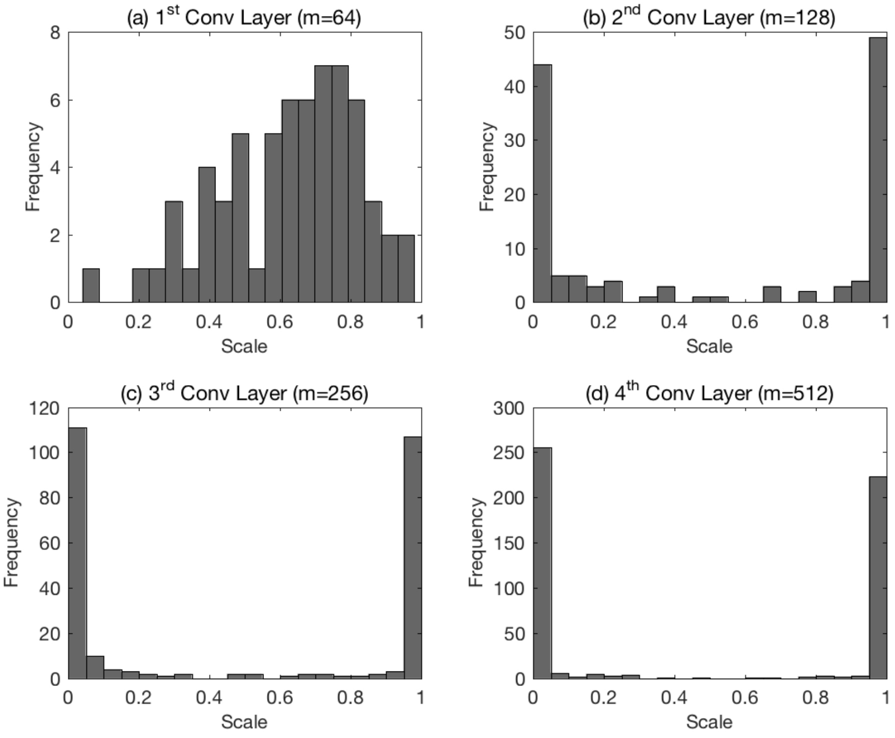


Figure 5.2: The histograms of scale values for the first four convolutional layers.

of parameters from the Scaled Module. This may be because the multiplication of scale values lessens the effect of noisy and redundant filters and improves the generalizability of the model. Filter pruning based on scale values lead to little loss in accuracy. With an increasing γ, the accuracy of both the Scaled Model and Scale-Pruned Model decreases, while the compression degree of the Scale-Pruned Model increases. More importantly, the Scale-Pruned Model achieved a much better performance than the Random-Pruned and l_1 norm-Pruned Baseline models when the same amount of filters in each layer was removed.

Figure 5.2 gives an example of the distributions of scale values for filters in each convolutional layer. Notably, most of the elements in $\mathbf{scale}_1$ stay around the initial value 0.73, while elements in $\mathbf{scale}_3$ and $\mathbf{scale}_4$ are either close to zero or one. It indicates that the filters in the first layer tend to have similar importance, which is in accordance with the findings in many publications [154, 155] that the first convo-

Method	Drowsiness F1-score (%)	Nondrowsiness F1-score (%)	Accuracy (%)
Scaled Model	76.46	73.15	75.02
Scale-Pruned Model	76.55	73.22	75.10
Baseline	74.55	72.02	73.53
l_1 norm-Pruned	73.26	70.56	72.21
Random-Pruned	66.84	63.75	65.79

Table 5.2: F1 score and accuracy on the evaluation set. $\gamma = 10^{-3}$ was used for the Scaled Model

lutional layer in a CNN extracts low-level features. The distribution of scale values in the next three layers indicates the existence of redundant filters. Note also that the percentage of redundant filters increases with the total number of filters in the convolutional layer.

Finally, based on the above results, a 3D CNN integrated with the Scale Module using $\gamma = 10^{-3}$ was trained. We then removed all filters with scale values < 0.5. Table 5.2 lists the average F1 scores and accuracies on the evaluation set from different models. The parameter reduction and FLOPs reduction in the Scale-Pruned model are 52.4% and 54.8%, respectively. The results show that there is no loss in accuracy after we removed over 50% of the filters. Also, a more than 4% improvement can be achieved through temporal smoothing with $L = 150$.

5.1.5 Summary and future work

In this study, we developed a drowsiness detection system and proposed a Scale Module to perform filter pruning. Our results show that our system can achieve good performance, and that the Scale Module can help us compress the CNN efficiently and reduce inference time. In our framework, redundant filters with small scale values can be removed after the model is trained with negligible effect on the accuracy, negating the need for further fine-tuning. Also, the Scale Module can be easily adapted to any state-of-the-art CNN structure and combined with other filter pruning techniques.

For future work, we will integrate the Scale Module into other state-of-art networks to further evaluate its performance.

ACKNOWLEDGMENT

This work is funded by DENSO International America, Inc.

5.2 Active Learning Framework

5.2.1 Introduction

One challenge of applying ML/DL in the medical domain is although there are plenty of medical images available, annotating these images is very time-consuming. To overcome the shortage of labeled images, an active learning strategy is presented. It started with training an initial SVM model using one annotated image and then queried the most informative superpixels whose labels may lead to the greatest improvement to the model. In this work, we will apply the proposed active learning framework in hematoma segmentation on brain CT scans from patients with TBI. The CT scans are over-segmented into superpixels. With active learning, a superpixel-based SVM classification model can be trained, which will result in a coarse hematoma segmentation. After that, an active contour model can be used to generate the final fine segmentation. Our experiments show that the active learning strategy can effectively select the most informative samples whose labels result in a significantly higher performance improvement compared with random selection. From our results, active learning can help overcome the shortage of annotated data, which is a common problem in medicine.

5.2.2 Dataset

Our dataset consists of 35 head CT scans from ProTECT III clinical trial [46] and 27 brain CT scans from the University of Michigan Health System. The brain scans are from patients who experienced a moderate to severe head injury and were enrolled in an emergency department within 4 hours of their injury. In total, 2433 axial CT images from 62 patients who suffered from acute TBI were used in this study, with image slice thickness ranging from 3.0 to 5.0 mm. To validation our proposed hematoma segmentation framework and active learning strategy, 13 cases were annotated as the test set by an experienced medical expert, who examined 2D cross-sectional slices and then manually drew the boundary around hematoma regions. We used the remaining 49 cases as the training set, wherein each experiment of active learning one slice was randomly selected and annotated as the initial training set while all others were used as the pool set.

5.2.3 Methods

5.2.3.1 Pre-processing

The CT scans underwent the same pre-processing steps as discussed in §2.4.1. In addition, after the contrast adjustment, a skull stripping method described in [56] was followed to extract brain tissues. For each CT slice, a rectangular contour was initialized around the center of the head. Then, the distance regularized level set evolution algorithm [156] was used to evolve the initialized contour to fit the border of the brain region enclosed by the skull. An example of contrast adjustment and skull stripping is shown in Figure 5.3 (a)-(b). The image after skull stripping is denoted as I_b.

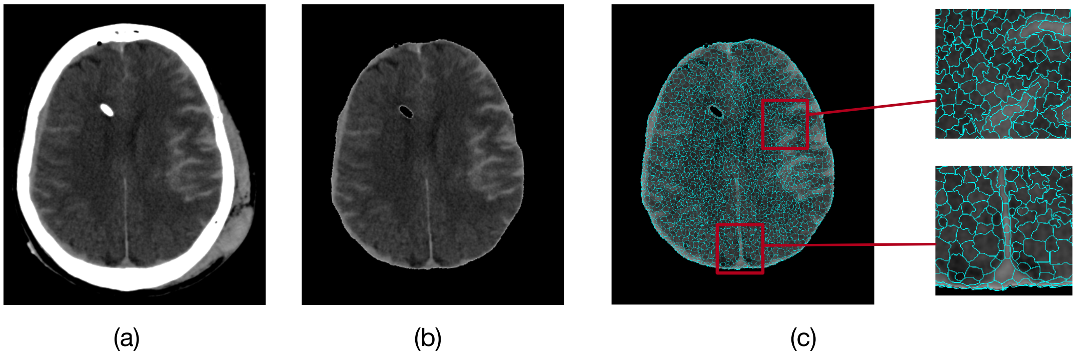


Figure 5.3: An illustration of the proposed data pre-processing and superpixel generation. (a) The image after the contrast adjustment. (b) The image after the skull stripping. (c) Superpixel generation.

5.2.3.2 Superpixel Generation

After pre-processing, we used the simple linear iterative clustering (SLIC) algorithm [157] to over-segment I_b into superpixels. The SLIC algorithm generates a group of coherent pixel collections based on color and spatial proximity, shown in Figure 5.3 (c). There are many advantages of using superpixels. First, instead of processing every image pixel, using superpixels where similar pixels are clustered can reduce computation cost efficiently. Secondly, superpixels divide the entire image into meaningful image patches. Features extracted from superpixels can better characterize regional information. Considering that superpixels adhere to edges within an image, as exhibited in Figure 5.3 (c), image segmentation can be performed via superpixel classification. In this work, we performed feature extraction on superpixels and classified those superpixels as belonging to hematoma regions or not. Based on these classification results, a coarse hematoma segmentation can be generated. In this study, I_b was over-segmented into approximately 5000 superpixels (each superpixel includes approximately 30 pixels).

5.2.3.3 Feature extraction

A total of 63 features were extracted to describe superpixels.

The mean, variance, skewness, and kurtosis of intensities in each superpixel were calculated. The mean value measures the average intensity level, while the variance measures heterogeneity. The skewness and kurtosis describe the asymmetry and the tailedness, respectively. As different ranges of Hounsfield units correspond to different anatomical structures, these intensity statistics can help to describe superpixels.

2-D Gabor filters oriented at 0, 30, 60, 90, 120, and 150 degrees with wavelengths of $2\sqrt{2}, 4\sqrt{2}, 8\sqrt{2}$, and $16\sqrt{2}$ were used to calculate the response map at $\gamma = 0.5$, $\psi = 0$ and $\sigma = 0.5\lambda$. The mean and variance of Gabor responses at each superpixel were calculated as Gabor features as well as the dominant spatial frequency and its orientation.

Saliency can be constructed as visual attention. A low-level approach was employed to determine the saliency of a superpixel by computing the average Euclidean distance of its mean intensity with 50 other superpixels that were randomly selected from the same image. Different from other extracted features, the saliency value contains global information at the slice level. From our observation, CT slices located close to the top of the head have a higher intensity value due to the partial volume effect. The saliency measurement can suppress the effect from this slice-level intensity shift. Also, it can help reduce the variability in intensity for the same tissue across different cases.

A 16×16 patch around the center of each superpixel was taken to calculate the gray-level co-occurrence matrix (GLCM), which gives the joint probability distribution of gray-level pairs of neighboring pixels. Let $\Omega_p = \{1, 2, \ldots, N_{level}\} \times \{1, 2, \ldots, N_{level}\}$, where N_{level} is the number of levels that gray intensities were quantized into. In this study, $N_{level} = 8$. Second-order statistics of the GLCM were used as features, specifically contrast, energy, and homogeneity, which are calculated as

In addition, a two-level discrete Haar wavelet packet transformation [158] was applied to a 16×16 patch around the center of each superpixel. The image patch was decomposed into 8 bands, with each band containing information of different frequencies. The energy of coefficients in each band was computed and the percentages of energy corresponding to the details were used as regional features to characterize each superpixel.

5.2.3.4 Active Learning

Active learning is a method [159] to train a supervised classifier with the smallest annotated training dataset possible. The proposed active learning framework is summarized in Figure 5.4. The active learning strategy started with training an initial SVM model using the initial training dataset, which consists of superpixels from only one labeled CT scan. After that, the initial model was used to classify superpixels from the pool dataset, which contains CT scans from 49 patients. Based on the predicted possibilities, we calculated the conditional Shannon entropy of each superpixel as

$$H_\Theta(s_i) = -\sum_{\hat{y}\in\{0,1\}} p_\Theta(y_i = \hat{y}|\boldsymbol{v}_i)\log(p_\Theta(y_i = \hat{y}|\boldsymbol{v}_i)), \tag{5.7}$$

where Θ denotes the trained SVM model. $\boldsymbol{v}_i$ and y_i are the feature vector and label of s_i, respectively. $p_\Theta(y_i = \hat{y}|v)$ denotes the predicted probability that s_i belongs to the corresponding class.

$H_\Theta(s_i)$ is used as an uncertainty measurement for s_i. A high $H_\Theta(s_i)$ indicates that the trained model is uncertain about which class s_i belongs to. This may occur if s_i is under-represented in the current training dataset. Thus superpixels with high uncertainty values are the most informative samples to update the model. In our work, superpixels were ranked based on their uncertainty measurements in descending order and the top N_{al} superpixels were selected to be annotated and added into the training dataset. Next, an updated SVM model was trained and the uncertainty

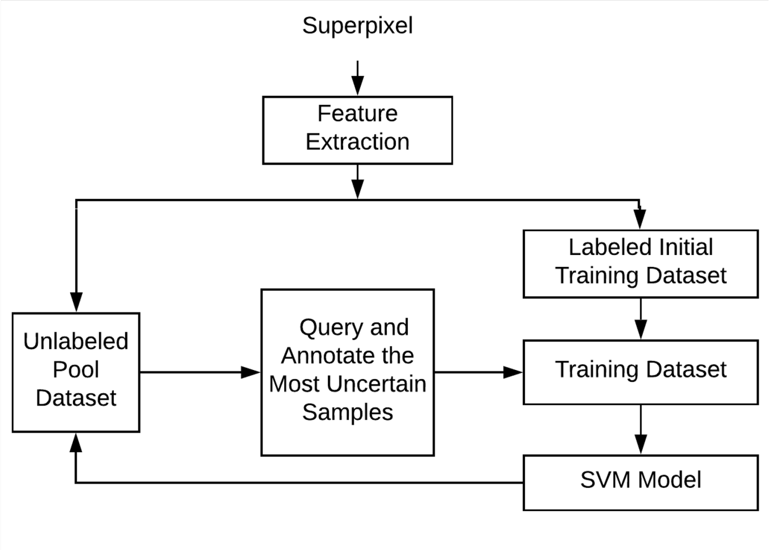

Superpixel
Feature Extraction
Labeled Initial Training Dataset
Unlabeled Pool Dataset
Query and Annotate the Most Uncertain Samples
Training Dataset
SVM Model

Figure 5.4: An overview of the proposed active learning framework.

measurements of the superpixels in the pool dataset were re-calculated. After the final SVM classifier was trained, coarse hematoma segmentation maps were generated by classifying superpixels in brain images.

5.2.4 Results and discussion

An initial SVM model was trained on the initial training set using a linear kernel. We then used the active learning strategy to select the most informative superpixels, which were then annotated, gradually improving the performance of the model. Several experiments were performed to explore the performance of classifiers with the same initial training set while varying N_{al} . From Figure 5.5, the curves tend to plateau after 1000 newly labeled superpixels are added with active learning, and the effect of N_{al} on the final performance is not significant. In contrast to the active learning strategy, an SVM classifier was trained as a baseline on the initial training set and a fixed number of randomly selected superpixels from the pool set (shown as "Random Selection" in Figure 5.5). 50 independent experiments were performed and the results were averaged to represent the performance of using random selection. From Figure 5.5, with the same number of added superpixels, the performance of the

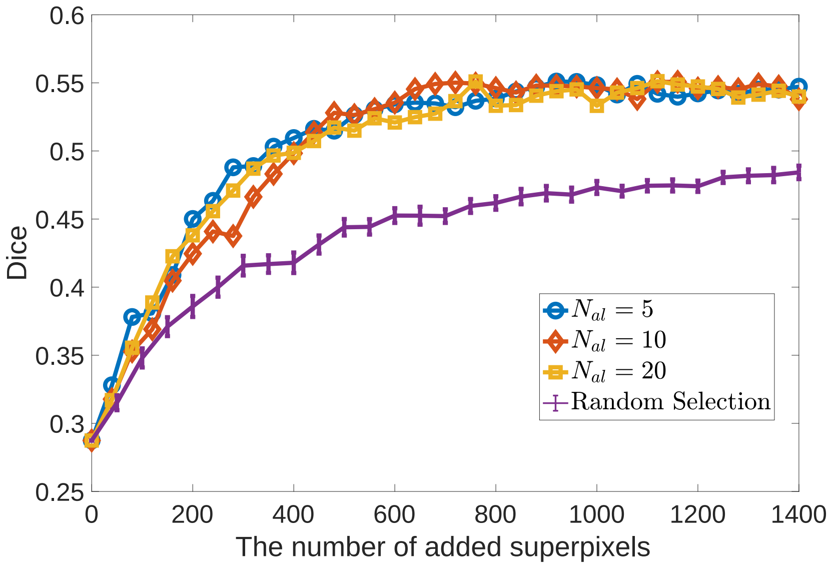


Figure 5.5: Comparison of active learning with $N_{al} = 5, 10, 20$ and random selection. The Dice coefficients of random selection with different numbers of added superpixels are averaged over 50 experiments. 95% confidence intervals are also given.

SVM model trained using the active learning strategy is significantly higher than the baseline. After adding over 1000 additional samples, the model trained with active learning achieved an average Dice coefficient of 0.55 over 13 patients.

To further examine the robustness of the active learning algorithm over different initial datasets, we repeated the above active learning algorithm and random selection method 20 times, respectively. For each time, one slice was randomly selected and annotated as the initial training set, while other slices in the training set were used as the pool set. Each SVM classifier using active learning was trained on the initial training set and 1000 additional samples selected with $N_{al} = 5$. The comparison of performance metrics between active learning and random selection is shown in Table 5.3. Our final SVM models from active learning have comparable performance with random selection models trained on the same initial training sets added with five times more annotated superpixels. The standard derivations of measures over experiments are very small.

After constructing the coarse hematoma segmentation for each slice via a trained

Model	Dice	Precision	Recall	Accuracy
Active Learning ($n = 1000$)	0.55 (0.01)	0.59 (0.02)	0.60 (0.02)	0.97 (0.01)
Random Selection ($n = 1000$)	0.47 (0.02)	0.45 (0.02)	0.61 (0.03)	0.94 (0.01)
Random Selection ($n = 5000$)	0.54 (0.01)	0.57 (0.02)	0.60 (0.02)	0.96 (0.01)

Table 5.3: Comparison of the active learning strategy and random selection method using 20 different initial training sets. n is the number of additional superpixels added to the initial training set. The mean and standard derivation (stddev) of evaluation measurements over 20 experiments are given in the format of mean (stddev).

SVM classifier, an active contour model [156] was used to refine the boundary. The coarse segmentation from the SVM classifier has an accurate hematoma localization while the boundary is rough, which may be due to superpixel sampling. The active contour model can help smooth the boundary and improve segmentation accuracy. From our result, the active contour model improved the segmentation performance by 5% in Dice coefficient.

CHAPTER VI

Conclusion

6.1 Summary

In this dissertation, I developed several CDS systems based on medical image/video processing and ML. With practical medical applications, novel mythologies that detect and recognize abnormalities or ROIs from medical images/videos were proposed, and quantitative features were calculated to capture characteristics of the patient's condition that are challenging to be collected by human reviewers. Novel strategies were proposed to improve the generalizability and interpretability of ML/DL models and overcome specific challenges in individual applications.

6.1.1 Quantitative hematoma evaluation and CDS System for patients with TBI

A CDS system was developed, which consists of brain hematoma segmentation, quantitative hematoma feature calculation, and 6-month mortality prediction. The 6-month mortality prediction model only uses information from admission data. It can help clinicians initially assess the severity and prognosis of a patient with TBI. The early outcome prediction can provide a reference for assessing the quality of health-care delivery and also help the clinical trial design, where patients may have a very good or a very poor prognosis based on the admission data can be excluded [160].

In this study, a novel Multi-view convolutional neural network with a mixed loss function was proposed for hematoma segmentation in head CT scans collected within 24 hours after injury. The proposed segmentation method was trained and validated on a clinical trial dataset of CT scans acquired using varying imaging protocols. The proposed hematoma segmentation network achieved an average Dice coefficient of 0.697 on the test set. Compared with other published methods on the same test set, the proposed method has the most accurate segmentation performance and volume estimation. In clinical settings, the proposed automatic hematoma segmentation method can reduce the time and workload of radiologists performing image segmentation and evaluating brain hematoma. The automated method is fast – segmenting hematoma in one series of CT scans takes less than 30 seconds. As such, an automated hematoma segmentation method can be used to evaluate hematoma expansion by processing a patient's CT scans at different time points. The automated method can reduce inter-observer and intra-observer variability. In addition, from our experimental results, the model has a great generalization on CT scans from multiple medical centers.

Based on the automated segmentation, a novel feature representation was proposed by calculating hematoma volume distribution in each anatomical region and shape features from hematoma segmentation results on brain CT scans. The proposed feature representation was combined with other clinical observations to predict 6-month mortality. The extracted volume and shape features' predictive power was explored and compared using 10-fold cross-validation on a clinical trial dataset consisting of 828 patients. The experimental results showed that CT-related features could significantly improve the 6-month mortality prediction. Extracted quantitative volume and shape features from the automated segmentation can better characterize the hematomas than manually evaluated qualitative and semi-quantitative features. Finally, the combination of “IMPACT without CT features” and extracted volume features led to a random forest model with an average AUPRC of 0.559 and AUC of 0.853, which are more than 10% and 5% higher than those of the widely used IMPACT model, respectively.

The work can be further improved by detecting hematoma subtypes and investigating how different types of hematoma contribute to outcome prediction. Additionally, the proposed mortality prediction model is tested using patient data from the PROTECT III trial, where patients with hypotension and severe hypoxemia were excluded from enrollment. In future work, other external datasets will be used to investigate whether the proposed model will generalize as well to the overall population of patients with moderate and severe TBI.

6.1.2 Colonoscopy video-based CDS system for patients with UC

A novel colonoscopy video-based CDS system was proposed for patients with UC. The system can estimate the disease severity and relative location for individual frames to derive a spatial severity distribution over the entire colon. Features were extracted from the severity distribution and achieved a good performance in MES estimation and outcome prediction. The automated system would provide broad accessibility to unbiased and reproducible disease assessments. Compared to central review costs of hundreds to thousands of dollars for each endoscopic video, automated computational scoring approaches are likely to provide a more cost-effective means for therapeutic trials, research, and clinical practice.

The camera localization module is a critical and challenging component of the proposed system. The localization system starts with the removal of non-informative frames and those containing biopsy forceps. The remaining frames are then fed into the motion estimation network to estimate camera motion between consecutive frames. The network is self-trained and does not require ground truth annotation. After that, the camera trajectory is derived, from which the location index is estimated. Based on the location index, a colon template was constructed by manually annotating times the camera entered each colon segment. With the estimated location index and colon template built from the training data, anatomical colon segment classification can be performed on a new colonoscopy video. The algorithm was trained using colonoscopy videos from routine practice. The motion estimation network's performance was validated on an external dataset, with the results showing that the proposed method is more accurate than other published methods. The proposed localization system was also validated using colonoscopy videos from routine practice. The performance of the colon segment classification was calculated and compared with baselines using either the time index or ScopeGuide length. The results show that using a camera-based location index achieves the best performance in colon segment classification. Additionally, we compared the trajectories of the camera motion-based location index and the ScopeGuide length-based location index. Similar patterns can be observed when there is no sharp spike or bump in ScopeGuide length-based location index sequences. These results indicate that the proposed localization system can accurately determine location awareness, which is critical in automated colonoscopy video analysis. The output of the localization system - the location index and anatomical colon segment classification - can facilitate contextual understanding in automated colonoscopy video analysis.

The disease severity distribution derived from the image analysis modules were used in MES estimation and outcome prediction. Based on our experimental results using videos collected from practical routine and clinical trials, the spatial disease severity distribution is more informative in evaluating a patient's condition and response to treatment. The proposed automated MES estimation algorithm has a good performance using high-quality endoscopic videos. The performance on external videos from a clinical trial is lower but still encouraging considering substantial variability in terms of (1) the video quality, compression, and color gamut; (2) the frequency of mucosal biopsies; (3) the duration of the colonoscopies.

The outcome prediction using the derived spatial severity distribution over the entire colon achieved a better performance than the model using humanly annotated MES and total score. It proves the predictive value from the spatial severity distribution. We will further investigate the spatial distribution of more comprehensive contextual information such as ulcers and erythema in our future work. Those features will better characterize the patient's condition and facilitate the decision-making models.

6.1.3 Tropical geometry-based interpretable ML algorithm and the application in patients with HF

Despite tremendous progress in the field of clinical decision support systems and AI/ML algorithms empowering such systems, there are still major challenges that prevent the widespread use of these methods in many similarly sensitive clinical applications. The challenges include (1) A wide spectrum of AI/ML methods are among the"black box" models whose use in clinical decision-making has been limited by a lack of transparency. Decision-makers in medicine are much less likely to trust recommendations for which no clear justification is provided. (2) In the majority of AI/ML models, there is no clear mechanism to leverage existing domain knowledge for model formation or training; (3) Many powerful methods such as DL utilize a large number of parameters, requiring tremendously large training and validation datasets. However, in many applications, generating large training datasets may be costly or even impossible.

To solve those challenges, a novel ML algorithm was proposed by the use of fuzzy logic and tropical geometry. The proposed network was tested on both synthetic datasets and a collected HF dataset. Our experimental results show that (1) The algorithm can learn hidden rules from the dataset and represent them in a way that humans can understand; (2) The introduction of the smoothness factor enables the algorithm to find the most suitable encoding functions and aggregation operators, which increases the performance of the proposed method; (3) Initializing the network with existing rough domain knowledge can effectively facilitate model training and improve the model's performance, especially when the size of the training set is limited.

In the HF application, the proposed method is applied to build a model that can identify patients with advanced HF who are appropriate for advanced therapies. The proposed network shows a significantly better generalizability compared with other existing ML techniques. Given the high prevalence of HF, the majority of patients are managed by primary care physicians or general cardiologists, who lack training in the management of patients with advanced HF. The proposed model can improve care to patients with HF.

6.1.4 Additional work

Additionally, a filter pruning technique was proposed to reduce the size of a trained DL model and speed up the inference phase. The redundancy in CNN will impair the model generalizability and accompanies unnecessary computation cost. With the proposed Scale Module, a network can estimate the importance of filters in a CNN and automatically reduce the weights of redundant filters in the training phase. The filter pruning technique can be applied to any regular CNN architecture and facilitate building a more efficient network.

An active learning framework was proposed to select the most informative samples in an unlabeled data pool and iteratively improve the model's performance by annotating the selected samples. The framework can avoid bias in the sample's annotation. It can help to identify samples under-represented in the annotated dataset by estimating the model's uncertainty on those samples.

6.2 Conclusion and Future Direction

In this dissertation, several clinical CDS systems have been proposed and validated on data collected from clinical practice or clinical trials. Major challenges in developing those AI-based systems and applying them in practice have been solved. The CDS system can provide clinicians with reproducible clinical measurements and recommendations in diagnosis, treatment, and outcome prediction. Ultimately, those systems can help improve patient management and the quality of life for patients and their caregivers.

In medical applications, it is critical to validate the proposed AI-based models using datasets with high variability. A strength of this dissertation is that the medical data collected or used are from multiple medical centers with different devices and imaging protocols. From our experimental results, our CDS systems have a good generalization of data from various sources. Our analysis also highlights the importance of digital data standardization.

This dissertation proposes novel algorithms to improve the generalizability and interpretability of AI/ML techniques. The proposed algorithms are transferable to other clinical or non-clinical applications. For example, the proposed interpretable ML algorithm with transparency and accessible reasoning can be easily utilized in other sensitive decision-making applications; the automated localization systems in colonoscopy video analysis can also help to provide location awareness to other endoscopic videos; the robust learning proposed for hematoma segmentation is also a general algorithm that helps to improve the DL model's invariance to image transformations.

In my future work, on the one hand, the decision-making models in the proposed

CDS systems will be further improved by integrating more clinical variables and validated on larger datasets with more diverse patient populations. On the other hand, the proposed algorithms will be applied to other medical applications to test their superiority.

APPENDICES

APPENDIX A

The association between individual features and the mortality

Table A.1: The association between features and the mortality.

Characteristics	Survival (n=676)	Mortality (n=152)	p-value
Age, years			<0.0001
<=30	310 (45.9)	30 (19.7)	
30-39	113 (16.7)	12 (7.9)	
40-49	107 (15.8)	16 (10.5)	
50-59	88 (13.1)	33 (21.7)	
60-69	38 (5.6)	24 (15.8)	
70+	20 (3.0)	37 (24.3)	
Best motor response, n (%)			0.00015
None/Extension	524 (77.5)	99 (65.1)	
Flexor response	77 (11.4)	28 (18.4)	
Withdrawal	39 (5.7)	21 (13.8)	
Localizes/obeys	36 (5.3)	4 (2.6)	
Pupillary reactivity, n (%)			<0.0001
Bilateral pupil response	91 (13.5)	34 (22.4)	
Unilateral pupil response	566 (83.7)	102 (67.1)	
No pupil response	19 (2.8)	16 (10.5)	
Marshall Score			<0.0001
I	104 (15.4)	3 (2.0)	
II	233 (34.5)	16 (10.5)	
III/IV/V/VI	339 (50.1)	133 (87.5)	
Traumatic subarachnoid hemorrhage, n (%)			0.00013
Yes	427 (63.2)	128 (84.2)	
Epidural hematoma, n (%)			0.018
Yes	89 (13.2)	32 (21.1)	
Glucose (mmol/l)			0.1
<6	99 (14.6)	13 (8.6)	
6-8.9	375 (55.5)	81 (53.3)	
9-11.9	150 (22.2)	44 (28.9)	
12-14.9	32 (4.7)	11 (7.2)	
15+	20 (3.0)	3 (2.0)	
Hb (g/dl)			<0.001
<9	11 (1.6)	4 (2.6)	
9-11.9	87 (12.9)	38 (25.0)	
12-14.9	416 (61.5)	88 (57.9)	
15+	162 (24.0)	22 (14.5)	
Total hematoma volumes			0.00042
Mean (std)	11.4 (22.2)	40.4 (36.7)	
Median (25th-75th percentile)	1.3 (0.1-11.5)	34.3 (10.3 - 61.4)	
Hematoma volumes in frontal lobe			<0.0001
Mean (std)	4.5 (10.3)	15.2 (17.0)	
Median (25th-75th percentile)	0.27 (0 - 3	10.1 (1.5 - 25.4)	
Hematoma volumes in temporal lobe			<0.0001
Mean (std)	0.1 (0.6)	0.4 (1.3)	
Median (25th-75th percentile)	0 (0-0)	0 (0 - 0.1)	
Hematoma volumes in parietal lobe			<0.0001
Mean (std)	2.03 (5.8)	6.3 (8.7)	
Median (25th-75th percentile)	0.04 (0 - 0.7)	2.6 (0.2 - 8.0)	
Hematoma volumes in occipital lobe			<0.0001
Mean (std)	2.5 (8.2)	11.4 (21.2)	
Median (25th-75th percentile)	0.2 (0 - 1.1)	3.1 (0.7 - 9.4)	
Hematoma volumes in posterior fossa			<0.0001
Mean (std)	2.1 (5.5)	6.0 (7.4)	
Median (25th-75th percentile)	0.02 (0 - 1.0)	2.6 (0.07 - 9.8)	

APPENDIX B

Camera model and image distortion correction

In this study, the camera at the tip of the colonoscope has a fisheye lens (Olympus PCF-H190). The fisheye lens achieves a very wide viewing angle by projecting a point in the 3D world frame to a half-hemisphere and then mapping the point to an image plane.

Figure B.1 is a diagram of the camera imaging model. Figure B.1(a) illustrates how a world coordinate frame, camera coordinate frame, and image plane are defined. The world coordinate frame is an arbitrarily-defined 3D coordinate system. The camera coordinate system is a 3D coordinate system based on the camera's optical center. The image plane is a 2D coordinate system, where the frame is generated from the camera, and its origin is the intersection between the z-axis (optical axis) and the image plane.

As shown in Figure B.1(b), a point in the 3D world coordinate frame $[X_w, Y_w, Z_w]^\mathsf{T}$ can be transformed into the camera coordinate system using the extrinsic parameters

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = R \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} + T, \tag{B.1}$$

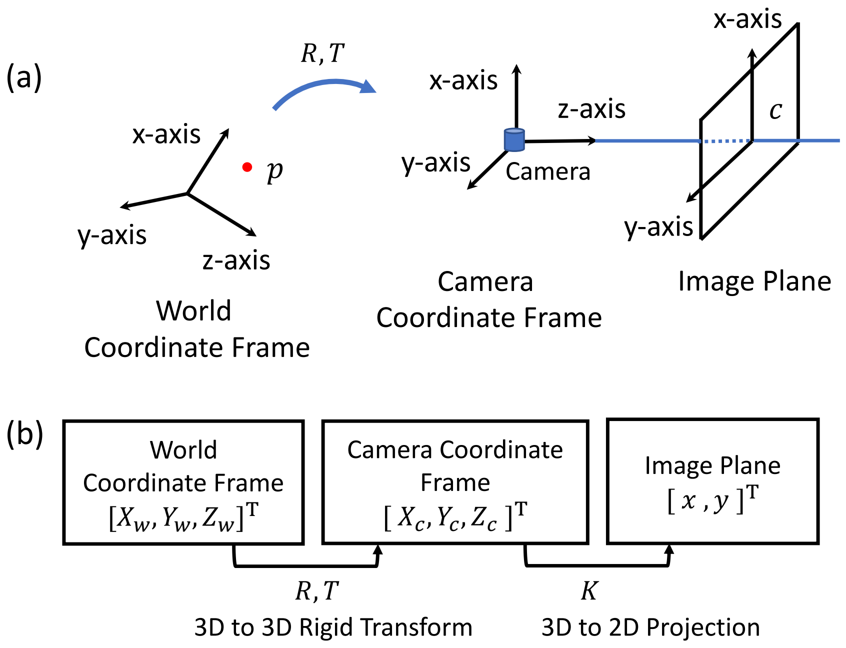


Figure B.1: Camera imaging model. (a) Diagram of each coordinate system. p is a point in the 3D world. c is the origin of the image plane (it is also called the image center); (b) A point in the world coordinate frame can be transformed into the camera coordinate frame via translation and rotation. The point can then be projected onto the image plane of the camera.

where R is a 3×3 rotation matrix and T is a 3×1 translation vector. R and T are extrinsic parameters of a camera that represent the camera's location in the world coordinates and the camera's orientation with respect to the world coordinate axes. If the world frame is the same as the camera frame, R is an identity matrix and T is the zero vector.

The resulting point $[X_c, Y_c, Z_c]^{\mathsf{T}}$ in the camera coordinate frame can be projected onto the image plane as $[x, y]^{\mathsf{T}}$ by the camera.

For a pinhole camera, the homogeneous coordinates $[x, y, 1]^{\mathsf{T}}$ of the projected point in the image plane can be written as

$$Z_c \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix}, \; K = \begin{pmatrix} f_x & e & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{B.2}$$

where K is called the intrinsic matrix of the camera, f_x and f_y are the focal lengths, x_0, y_0 are the principal point offsets, and e is the axis skew.

In this study, the mathematical model of a fisheye camera proposed in [111] was utilized. The mapping of camera coordinates $[X_c, Y_c, Z_c]^\mathsf{T}$ to the 2D pixel coordinates $[x, y]^\mathsf{T}$ in the image plane of a fisheye camera can be written as

$$\beta \begin{pmatrix} x - x_0 \\ y - y_0 \\ d(l) \end{pmatrix} = \beta \begin{pmatrix} x - x_0 \\ y - y_0 \\ \sum_{i=0}^{n} a_i l^i \end{pmatrix} = \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} \tag{B.3a}$$

$$d(l) = \sum_{i=0}^{n} a_i l^i \tag{B.3b}$$

$$l = \sqrt{(x - x_0)^2 + (y - y_0)^2}, \tag{B.3c}$$

where β is the scaling factor; $d(r)$ is the image distortion; a_0, a_1, ..., a_n are intrinsic parameters of the camera; and l is the distance from the image center $[x_0, y_0]^\mathsf{T}$.

With a fisheye lens the generated image will have a convex non-rectilinear appearance. Image distortion correction is essential for the subsequent optical flow calculation and camera motion estimation. For distortion correction, camera calibration is first performed to estimate the camera's intrinsic parameters a_0, ..., a_n, (n can also be determined during camera calibration), and K. In the process of camera calibration, a paper with a checkerboard pattern is placed in front of the fisheye camera and a number of frames from different angles are captured. After that, the camera's intrinsic parameters are estimated by calculating how straight lines in the checkerboard are distorted. After the camera calibration, the distorted pixel coordinates $[x, y]^\mathsf{T}$ can be converted to pixel coordinates $[x', y']^\mathsf{T}$ using equations (B.2) and

(B.3):

$$K^{-1}\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \beta \begin{pmatrix} x - x_0 \\ y - y_0 \\ d(r) \end{pmatrix}. \tag{B.4}$$

Based on equation (B.4), x and y can be written as

$$x' = \frac{f_x}{d(r)}(x - x_0) + \frac{e}{d(r)}(y - y_0) + x'_0, \tag{B.5a}$$

$$y' = \frac{f_y}{d(r)}(y - y_0) + y'_0, \tag{B.5b}$$

where $[x'_0, y'_0]^\mathsf{T}$ is the center of the corrected image. In this way, image distortion can be eliminated, and the corrected image can be regarded as an image captured by a pinhole camera with the estimated intrinsic matrix K.

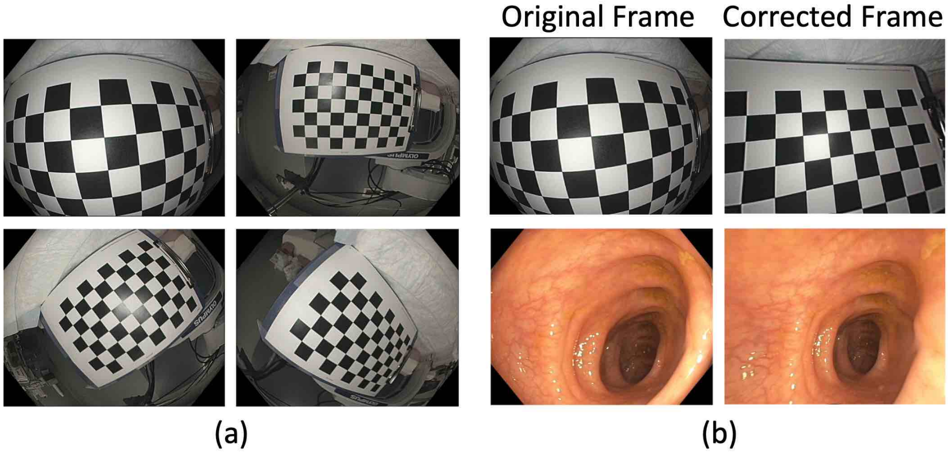


Figure B.2: Camera calibration and frame correction. (a) Examples of frames captured in camera calibration (b) Examples of image distortion correction.

Figure B.2 (a) presents several examples of frames captured for camera calibration. Frames of a checkerboard pattern were taken from different angles and distances to capture the characteristics of the fisheye lens. With the calculated intrinsic parameters, the distorted images can be corrected. Figure B.2 (b) shows two examples of distorted image correction. The first one is a frame of a checkerboard pattern,in which we can see the lines of the checkerboard pattern are straight after the original frame was corrected. The second one is a frame of the colon. All frames in colonoscopy videos from the localization dataset were corrected to remove distortion.

APPENDIX C

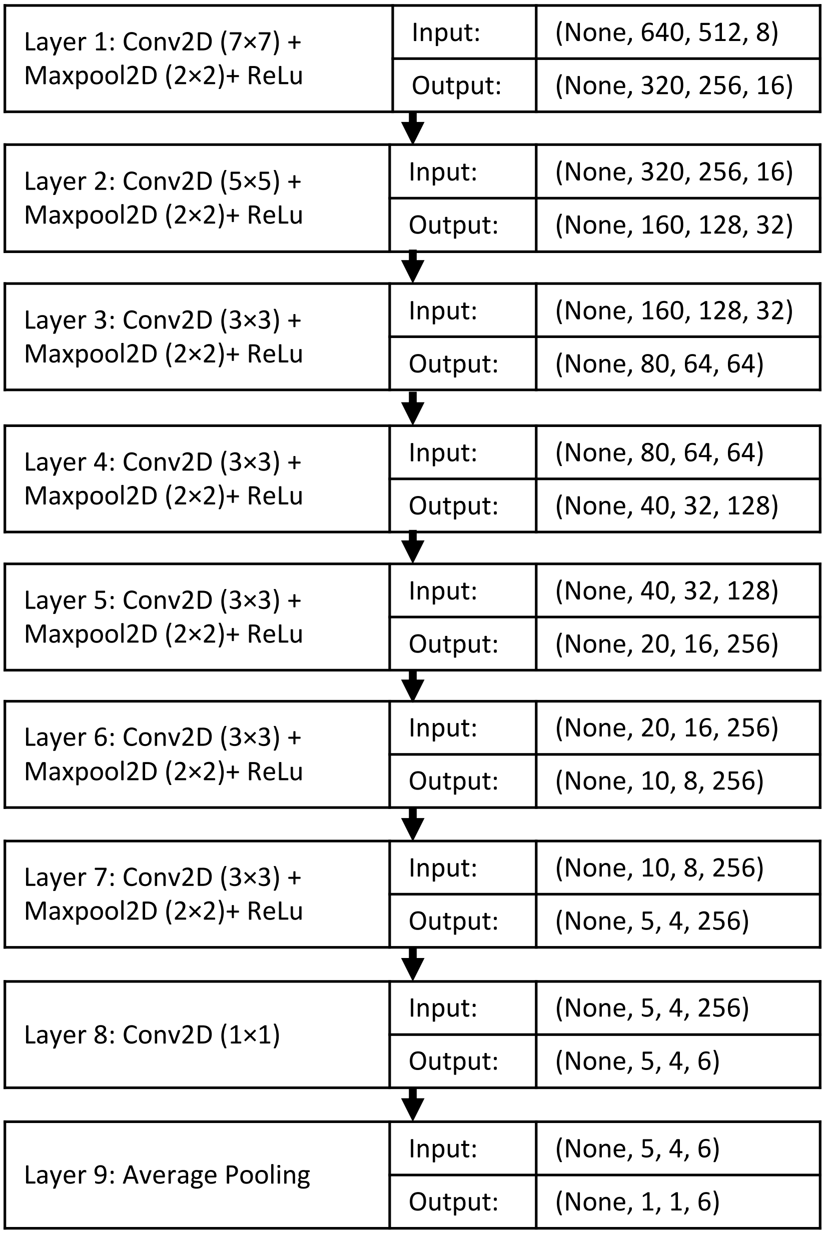
Optical flow calculation

The proposed method requires optical flow as part of input to the motion estimation network. In this study, a pre-trained PWC-Net model from [112] was used. The PWC-Net model was first trained on the FlyingChairs dataset consisting of 22,872 image pairs [161] and then fine-tuned on the FlyingThings3D dataset consisting of 35,000 image pairs [114]. In their published results, PWC-Net achieved good performance with respect to several benchmarks. A deep learning based optical flow calculation method was chosen considering the complex geometry and limited textural pattern across the colon. From [112], PWC-Net is robust to real images where the image edges are often corrupted by motion blur and noise. We also evaluated the built-in Lucas-Kanade optical flow method from OpenCV library. On the validation set of the localization dataset Set 1, the performance of the pose estimation decreased slightly.

APPENDIX D

Architectures of the motion network and disparity network

The detailed architectures of the motion network and disparity network are shown in Figure D.1 and Figure D.2. For each layer, the filter size, activation function, input size, and output size are given. In Figure D.1 and Figure D.2, “Conv2D” means 2D convolutional layer; “Maxpool2D” means 2D max pooling layer; “Transpose2D” means 2D transpose layer. The shape of the input and output for each block is presented in the format of (batch size, height, width, the number of channels). “None” means that the batch size is of arbitrary value.



Layer 1: Conv2D (7×7) + Maxpool2D (2×2)+ ReLu	Input:	(None, 640, 512, 8)
	Output:	(None, 320, 256, 16)
Layer 2: Conv2D (5×5) + Maxpool2D (2×2)+ ReLu	Input:	(None, 320, 256, 16)
	Output:	(None, 160, 128, 32)
Layer 3: Conv2D (3×3) + Maxpool2D (2×2)+ ReLu	Input:	(None, 160, 128, 32)
	Output:	(None, 80, 64, 64)
Layer 4: Conv2D (3×3) + Maxpool2D (2×2)+ ReLu	Input:	(None, 80, 64, 64)
	Output:	(None, 40, 32, 128)
Layer 5: Conv2D (3×3) + Maxpool2D (2×2)+ ReLu	Input:	(None, 40, 32, 128)
	Output:	(None, 20, 16, 256)
Layer 6: Conv2D (3×3) + Maxpool2D (2×2)+ ReLu	Input:	(None, 20, 16, 256)
	Output:	(None, 10, 8, 256)
Layer 7: Conv2D (3×3) + Maxpool2D (2×2)+ ReLu	Input:	(None, 10, 8, 256)
	Output:	(None, 5, 4, 256)
Layer 8: Conv2D (1×1)	Input:	(None, 5, 4, 256)
	Output:	(None, 5, 4, 6)
Layer 9: Average Pooling	Input:	(None, 5, 4, 6)
	Output:	(None, 1, 1, 6)

Figure D.1: The detailed architecture for the motion network.

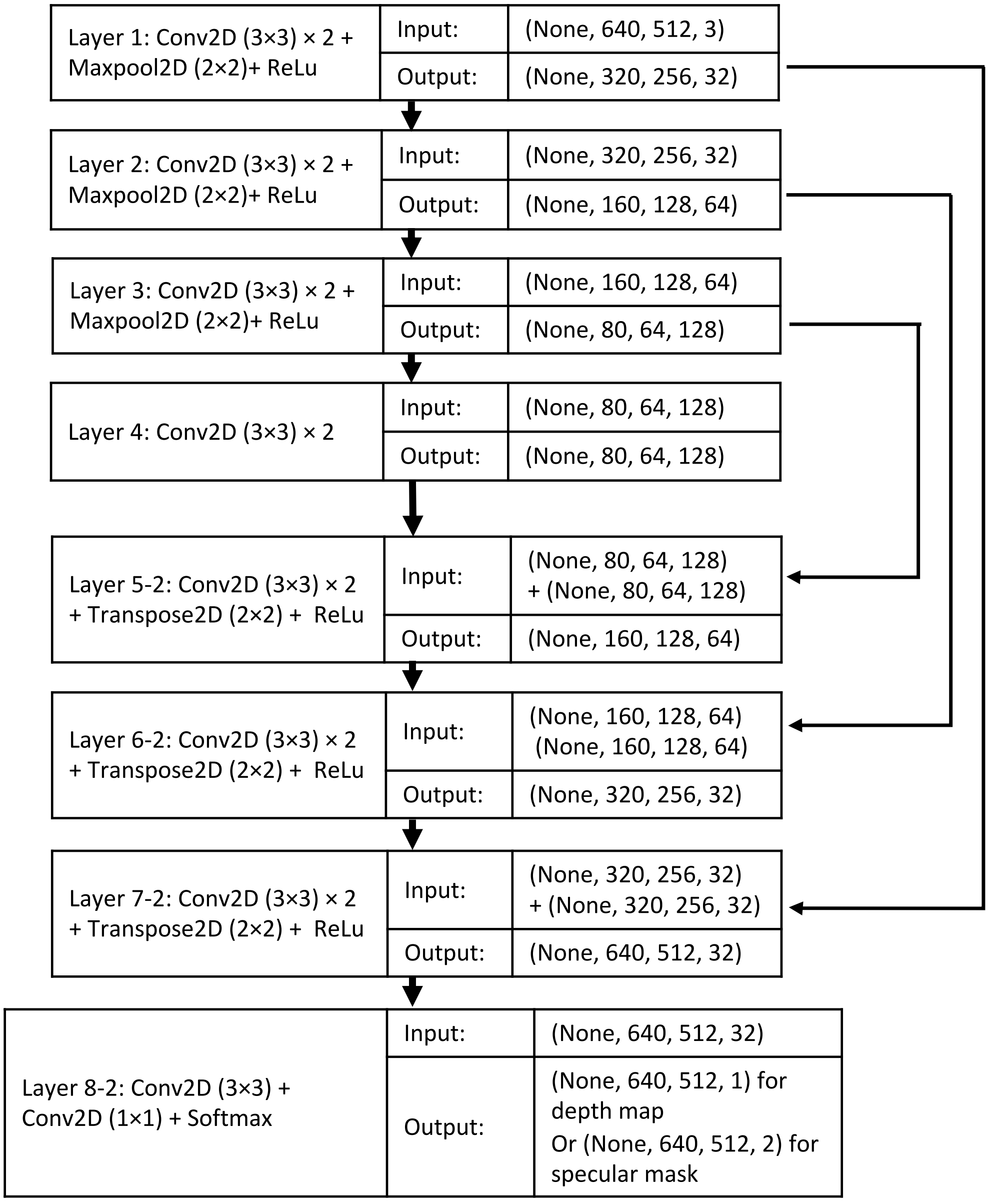

Layer 1: Conv2D (3×3) × 2 + Maxpool2D (2×2)+ ReLu
Input:
(None, 640, 512, 3)
Output:
(None, 320, 256, 32)
Layer 2: Conv2D (3×3) × 2 + Maxpool2D (2×2)+ ReLu
Input:
(None, 320, 256, 32)
Output:
(None, 160, 128, 64)
Layer 3: Conv2D (3×3) × 2 + Maxpool2D (2×2)+ ReLu
Input:
(None, 160, 128, 64)
Output:
(None, 80, 64, 128)
Layer 4: Conv2D (3×3) × 2
Input:
(None, 80, 64, 128)
Output:
(None, 80, 64, 128)
Layer 5-2: Conv2D (3×3) × 2 + Transpose2D (2×2) + ReLu
Input:
(None, 80, 64, 128) + (None, 80, 64, 128)
Output:
(None, 160, 128, 64)
Layer 6-2: Conv2D (3×3) × 2 + Transpose2D (2×2) + ReLu
Input:
(None, 160, 128, 64) (None, 160, 128, 64)
Output:
(None, 320, 256, 32)
Layer 7-2: Conv2D (3×3) × 2 + Transpose2D (2×2) + ReLu
Input:
(None, 320, 256, 32) + (None, 320, 256, 32)
Output:
(None, 640, 512, 32)
Layer 8-2: Conv2D (3×3) + Conv2D (1×1) + Softmax
Input:
(None, 640, 512, 32)
Output:
(None, 640, 512, 1) for depth map
Or (None, 640, 512, 2) for specular mask

Figure D.2: The detailed architecture for the disparity network.

APPENDIX E

Generic colon template and colon length variation

In this study, we estimated a patient-generic template for the lengths of the colon segments to perform anatomical colon segment classification. There are some other sources that can be used to estimate the colon length. For example, in [162], the colon segment length and variation across patients were analyzed for patients undergoing barium enema examination. CT or MR abdominal images can also be used to calculate a colon template. However, during a colonoscopy the colon will be stretched and the length of the colon segment will be different from the length under normal conditions. Additionally, the shapes of colon segments vary - the ascending and transverse colon are relatively straight, while the the sigmoid colon has an 'S' shape. Considering the motility of the colon, its extreme distensibility, and the impact of patient position (different for colonoscopy, CT, or MRI), the shape, length, and configuration of the colon changes during colonoscopy. As a result, the length of the colon segment after stretching cannot be inferred without a colonoscopy. Thus, CT or MR abdominal images cannot be used for colon segment template building. As the clinical importance of the localization algorithm is to link disease features on frames from colonoscopy video to location awareness, it is better to use the colon template built from colonoscopy videos.

	Cecum	Ascending colon	Transverse colon	Descending colon	Sigmoid colon	Rectum
Mean	0.067	0.142	0.245	0.204	0.244	0.097
25^{th} percentile	0.039	0.110	0.150	0.134	0.161	0.061
75^{th} percentile	0.087	0.171	0.336	0.284	0.306	0.103
Standard deviation	0.038	0.061	0.097	0.088	0.086	0.057

Table E.1: Average, range, and variation of the estimated length for each manually annotated colon segment on Set 3 of the localization set.

The colon template was constructed based on the assumption that the relative lengths of the colon segments are similar across patients. [162] shows the variation of absolute colon segment length across the patients. Although the definition of colon segments is slightly different and the length cannot be used for building the colon template directly for the aforementioned reasons, the variation can be used as a reference. From their results, the variations in the length of the rectum and cecum are small while the variations in the length of the ascending colon to the sigmoid colon are relatively larger (up to 10 cm). In Table E.1, the range and variation of the estimated length for each manually annotated colon segment on the independent test set (Set 3 of the localization set) are presented. Compared to the variation in [162], the variation shown in Table E.1 is close and slightly higher, which may be due to the errors from location index estimation and manually colon segment annotations. Though variation of the relative colon segment length across patients exists, the anatomical colon segment classification is clinically important. Considering the anatomical geometry and the motility and distensibility of the colon, building a patient-generic template from colonoscopy videos is the best way of proceeding compared to other alternatives without introducing additional sensors. Many clinical applications only need relative location and generalize anatomic spatial information. For example, information like "approximately 20% is affected and the region is near the sigmoid colon and rectum" can be very important to evaluate the patient's condition and predict outcomes.

APPENDIX F

Baseline using ScopeGuide length

Currently, ScopeGuide is the best clinical measurement tool with FDA approval that can be used to estimate the traveled distance in colonoscopy videos. ScopeGuide's principle use is as a training tool to inform physicians of the shape and configuration of the scope inside the patient, not as a positioning system. The raw data from the electromagnetic sensors utilized by ScopeGuide is not accessible. The distance measurement provided, though crude, offered the best objective localization that is currently available and FDA approved for in human use. Alternative methods would require real-time fluoroscopy (exposing the patient to ionizing radiation) or additional sensor-based approaches that are not FDA approved. Ideally, a specialized sensor would provide the best estimate of motion and spatial data, and its accuracy is posited to be superior to a vision-based system. However, vision-based camera localization methods are still attractive as there is no additional equipment needed, allowing for rapid integration into clinical workflows, wide availability, and low cost.

APPENDIX G

Implementation of the established ML algorithms

Public python packages were used to build the established ML classifiers with default settings except for specified hyper-parameters to be tuned on the validation set.

1. Logistic Regression: We used the Logistic Regression Classifier from *sklearn*[163].
2. Naïve Bayes: We used Gaussian Naive Bayes from *sklearn*.
3. Decision Tree: We used Decision Tree Classifier from *sklearn*. The maximal depth of the tree and the minimum number of samples required to split were tuned.
4. Random Forest: We used Random Forest Classifier from *sklearn*. The number of trees, the maximal depth of the tree, and the minimum number of samples required to split were tuned.
5. SVM: We used Support Vector Classifier from *sklearn*, whose implementation is based on *libsvm* [164]. The regularization parameter, the kernel type (linear function, radial basis function, sigmoid function, or polynomial function), and kernel coefficient were tuned.

6. XGBoost: We used the tree-based XGBoost Classifier from *xgboost* [165]. The number of boosting rounds, learning rate, maximal tree depth for base learners were tuned.

7. EBM: We used the Explainable Boosting Classifier from *interpret*. The learning rate and ways of feature interactions were tuned.

8. Fuzzy inference classifier: We used Fuzzy Reduction Rule from *fylearn*, which is the best one from our preliminary analysis on public ML benchmark datasets. The classifier used a pi-type membership function and fuzzy mean aggregation [148].

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Pavel Hamet and Johanne Tremblay. Artificial intelligence in medicine. *Metabolism*, 69:S36–S40, 2017.

[2] James H Thrall, Xiang Li, Quanzheng Li, Cinthia Cruz, Synho Do, Keith Dreyer, and James Brink. Artificial intelligence and machine learning in radiology: opportunities, challenges, pitfalls, and criteria for success. *Journal of the American College of Radiology*, 15(3):504–508, 2018.

[3] Linda Sundvall, Hans Jakob Ingerslev, Ulla Breth Knudsen, and Kirstine Kirkegaard. Inter-and intra-observer variability of time-lapse annotations. *Human Reproduction*, 28(12):3215–3221, 2013.

[4] A Plaza-Florido, JMA Alcantara, JH Migueles, FJ Amaro-Gahete, FM Acosta, J Mora-Gonzalez, J Sacha, and FB Ortega. Inter-and intra-researcher reproducibility of heart rate variability parameters in three human cohorts. *Scientific reports*, 10(1):1–11, 2020.

[5] John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric K Oermann. Confounding variables can degrade generalization performance of radiological deep learning models. *arXiv preprint arXiv:1807.00431*, 2018.

[6] Chinmay Belthangady and Loic A Royer. Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction. *Nature methods*, 16(12):1215–1225, 2019.

[7] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430, 2018.

[8] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[9] Gregor Stiglic, Primoz Kocbek, Nino Fijacko, Marinka Zitnik, Katrien Verbert, and Leona Cilar. Interpretability of machine learning-based prediction models in healthcare. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(5):e1379, 2020.

[10] Alfredo Vellido. The importance of interpretability and visualization in machine learning for applications in medicine and health care. *Neural computing and applications*, 32(24):18069–18083, 2020.

[11] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[12] David W Wright, Arthur Kellermann, Lisa C McGuire, Bin Chen, and Tanja Popovic. Cdc grand rounds: reducing severe traumatic brain injury in the united states. *MMWR. Morbidity and mortality weekly report*, 62(27):549, 2013.

[13] Victor G Coronado, Lisa C McGuire, Mark Faul, David E Sugerman, and William S Pearson. Traumatic brain injury epidemiology and public health issues. *Brain injury medicine: Principles and practice*, 84, 2012.

[14] Bruce Lee and Andrew Newberg. Neuroimaging in traumatic brain imaging. *NeuroRx*, 2(2):372–383, 2005.

[15] Jane J Kim and Alisa D Gean. Imaging for the diagnosis and management of traumatic brain injury. *Neurotherapeutics*, 8(1):39–53, 2011.

[16] Joseph P Broderick, Thomas G Brott, John E Duldner, Thomas Tomsick, and Gertrude Huster. Volume of intracerebral hemorrhage. a powerful and easy-to-use predictor of 30-day mortality. *Stroke*, 24(7):987–993, 1993.

[17] Bram Jacobs, Tjemme Beems, Ton M van der Vliet, Ramon R Diaz-Arrastia, George F Borm, and Pieter E Vos. Computed tomography and outcome in moderate and severe traumatic brain injury: hematoma volume and midline shift revisited. *Journal of neurotrauma*, 28(2):203–215, 2011.

[18] Chih-Wei Wang, Yi-Jui Liu, Yi-Hsiung Lee, Dueng-Yuan Hueng, Hueng-Chuen Fan, Fu-Chi Yang, Chun-Jen Hsueh, Hung-Wen Kao, Chun-Jung Juan, and Hsian-He Hsu. Hematoma shape, hematoma size, glasgow coma scale score and ich score: which predicts the 30-day mortality better for intracerebral hematoma? *PloS one*, 9(7):e102326, 2014.

[19] M Ross Bullock, Randall Chesnut, Jamshid Ghajar, David Gordon, Roger Hartl, David W Newell, Franco Servadei, Beverly C Walters, and Jack Wilberger. Surgical management of posterior fossa mass lesions. *Neurosurgery*, 58(suppl_3):S2–47, 2006.

[20] Randall Chesnut, Jamshid Ghajar, and David Gordon. Surgical management of acute epidural hematomas. *Neurosurgery*, 58(3):S2–7, 2006.

[21] Andrew F Ducruet, Zachary L Hickman, Brad E Zacharia, Bartosz T Grobelny, Peter A DeRosa, Elissa Landes, Shuang Lei, Joyce Khandji, Sarah Gutbrod,

and E Sander Connolly. Impact of platelet transfusion on hematoma expansion in patients receiving antiplatelet agents before intracerebral hemorrhage. *Neurological research*, 32(7):706–710, 2010.

[22] Hee-Kwon Park, Seung-Hoon Lee, Kon Chu, and Jae-Kyu Roh. Effects of celecoxib on volumes of hematoma and edema in patients with primary intracerebral hemorrhage. *Journal of the neurological sciences*, 279(1-2):43–46, 2009.

[23] KN Bhanu Prakash, Shi Zhou, Tim C Morgan, Daniel F Hanley, and Wieslaw L Nowinski. Segmentation and quantification of intra-ventricular/cerebral hemorrhage in ct scans by modified distance regularized level set evolution technique. *International journal of computer assisted radiology and surgery*, 7(5):785–798, 2012.

[24] Rashmi U Kothari, Thomas Brott, Joseph P Broderick, William G Barsan, Laura R Sauerbeck, Mario Zuccarello, and Jane Khoury. The abcs of measuring intracerebral hemorrhage volumes. *Stroke*, 27(8):1304–1305, 1996.

[25] Chih-Wei Wang, Chun-Jung Juan, Yi-Jui Liu, Hsian-He Hsu, Hua-Shan Liu, Cheng-Yu Chen, Chun-Jen Hsueh, Chung-Ping Lo, Hung-Wen Kao, and Guo-Shu Huang. Volume-dependent overestimation of spontaneous intracerebral hematoma volume by the abc/2 formula. *Acta Radiologica*, 50(3):306–311, 2009.

[26] Hagen B Huttner, Thorsten Steiner, Marius Hartmann, Martin Köhrmann, Eric Juettler, Stephan Mueller, Johannes Wikner, Uta Meyding-Lamade, Peter Schramm, Stefan Schwab, et al. Comparison of abc/2 estimation technique to computer-assisted planimetric analysis in warfarin-related intracerebral parenchymal hemorrhage. *Stroke*, 37(2):404–408, 2006.

[27] Chun-Chih Liao, Furen Xiao, Jau-Min Wong, and I-Jen Chiang. Computer-aided diagnosis of intracranial hematoma with brain deformation on computed tomography. *Computerized medical imaging and graphics*, 34(7):563–571, 2010.

[28] HS Bhadauria and ML Dewal. Intracranial hemorrhage detection using spatial fuzzy c-mean and region-based active contour on brain ct imaging. *Signal, Image and Video Processing*, 8(2):357–364, 2014.

[29] Manas Kumar Nag, Saunak Chatterjee, Anup Kumar Sadhu, Jyotirmoy Chatterjee, and Nirmalya Ghosh. Computer-assisted delineation of hematoma from ct volume using autoencoder and chan vese model. *International journal of computer assisted radiology and surgery*, 14(2):259–269, 2019.

[30] Saurabh Jain, Thijs Vande Vyvere, Vasilis Terzopoulos, Diana Maria Sima, Eloy Roura, Andrew Maas, Guido Wilms, and Jan Verheyden. Automatic quantification of computed tomography features in acute traumatic brain injury. *Journal of neurotrauma*, 36(11):1794–1803, 2019.

[31] PD Chang, E Kuoy, J Grinband, BD Weinberg, M Thompson, R Homo, J Chen, H Abcede, M Shafie, L Sugrue, et al. Hybrid 3d/2d convolutional neural network for hemorrhage evaluation on head ct. *American Journal of Neuroradiology*, 39(9):1609–1616, 2018.

[32] Ewout W Steyerberg, Nino Mushkudiani, Pablo Perel, Isabella Butcher, Juan Lu, Gillian S McHugh, Gordon D Murray, Anthony Marmarou, Ian Roberts, J Dik F Habbema, et al. Predicting outcome after traumatic brain injury: development and international validation of prognostic scores based on admission characteristics. *PLoS medicine*, 5(8):e165, 2008.

[33] S Mohamad R Soroushmehr, A Bafna, S Schlosser, Kevin Ward, Harm Derksen, and Kayvan Najarian. Ct image segmentation in traumatic brain injury. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2973–2976. IEEE, 2015.

[34] Soumi Ray, Vinod Kumar, Chirag Ahuja, and Niranjan Khandelwal. Intensity population based unsupervised hemorrhage segmentation from brain ct images. *Expert Systems with Applications*, 97:325–335, 2018.

[35] Snehashis Roy, Sean Wilkes, Ramon Diaz-Arrastia, John A Butman, and Dzung L Pham. Intraparenchymal hemorrhage segmentation from clinical head ct of patients with traumatic brain injury. In *Medical Imaging 2015: Image Processing*, volume 9413, page 94130I. International Society for Optics and Photonics, 2015.

[36] Wei Tu, Linglong Kong, Rohana Karunamuni, Ken Butcher, Lili Zheng, and Rebecca McCourt. Nonlocal spatial clustering in automated brain hematoma and edema segmentation. *Applied Stochastic Models in Business and Industry*, 35(2):321–329, 2019.

[37] Pankaj Singh, Vandana Khanna, and Meenu Kamal. Hemorrhage segmentation by fuzzy c-mean with modified level set on ct imaging. In *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 550–555. IEEE, 2018.

[38] Indrajeet Kumar, Chandradeep Bhatt, and Kamred Udham Singh. Entropy based automatic unsupervised brain intracranial hemorrhage segmentation using ct images. *Journal of King Saud University-Computer and Information Sciences*, 2020.

[39] Justin L Wang, Hassan Farooq, Hanqi Zhuang, and Ali K Ibrahim. Segmentation of intracranial hemorrhage using semi-supervised multi-task attention-based u-net. *Applied Sciences*, 10(9):3297, 2020.

[40] Ryan T Kellogg, Jan Vargas, Guilherme Barros, Rajeev Sen, David Bass, J Ryan Mason, and Michael Levitt. Segmentation of chronic subdural

hematomas using 3d convolutional neural networks. *World Neurosurgery*, 148:e58–e65, 2021.

[41] Ajay Patel, Floris HBM Schreuder, Catharina JM Klijn, Mathias Prokop, Bram van Ginneken, Henk A Marquering, Yvo BWEM Roos, M Irem Baharoglu, Frederick JA Meijer, and Rashindra Manniesing. Intracerebral haemorrhage segmentation in non-contrast ct. *Scientific reports*, 9(1):1–11, 2019.

[42] Bram Jacobs, Tjemme Beems, Ton M van der Vliet, Arie B van Vugt, Cornelia Hoedemaekers, Janneke Horn, Gaby Franschman, Ian Haitsma, Joukje van der Naalt, Teuntje MJC Andriessen, et al. Outcome prediction in moderate and severe traumatic brain injury: a focus on computed tomography variables. *Neurocritical care*, 19(1):79–89, 2013.

[43] Lawrence F Marshall, Sharon Bowers Marshall, Melville R Klauber, Marjan van Berkum Clark, Howard M Eisenberg, John A Jane, Thomas G Luerssen, Anthony Marmarou, and Mary A Foulkes. A new classification of head injury based on computerized tomography. *Journal of neurosurgery*, 75(Supplement):S14–S20, 1991.

[44] Andrew IR Maas, Chantal WPM Hukkelhoven, Lawrence F Marshall, and Ewout W Steyerberg. Prediction of outcome in traumatic brain injury with computed tomographic characteristics: a comparison between the computed tomographic classification and combinations of computed tomographic predictors. *Neurosurgery*, 57(6):1173–1182, 2005.

[45] Esther L Yuh, Shelly R Cooper, Adam R Ferguson, and Geoffrey T Manley. Quantitative ct improves outcome prediction in acute traumatic brain injury. *Journal of neurotrauma*, 29(5):735–746, 2012.

[46] David W Wright, Sharon D Yeatts, Robert Silbergleit, Yuko Y Palesch, Vicki S Hertzberg, Michael Frankel, Felicia C Goldstein, Angela F Caveney, Harriet Howlett-Smith, Erin M Bengelink, et al. Very early administration of progesterone for acute traumatic brain injury. *New England Journal of Medicine*, 371(26):2457–2466, 2014.

[47] Karel Zuiderveld and Paul S Heckbert. Graphics gems iv. *San Diego, CA, USA: Academic Press Professional, Inc*, pages 474–485, 1994.

[48] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[49] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[50] Wei Shen, Mu Zhou, Feng Yang, Caiyun Yang, and Jie Tian. Multi-scale convolutional neural networks for lung nodule classification. In *International Conference on Information Processing in Medical Imaging*, pages 588–599. Springer, 2015.

[51] Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78, 2017.

[52] Saurabh Sharma. *Analysis of Stroke on Brain Computed Tomography Scans*. PhD thesis, International Institute of Information Technology Hyderabad, 2013.

[53] Stefan Klein, Marius Staring, Keelin Murphy, Max A Viergever, and Josien PW Pluim. Elastix: a toolbox for intensity-based medical image registration. *IEEE transactions on medical imaging*, 29(1):196–205, 2009.

[54] Mobarakol Islam, Parita Sanghani, Angela An Qi See, Michael Lucas James, Nicolas Kon Kam King, and Hongliang Ren. ICHnet: Intracerebral hemorrhage (ICH) segmentation using deep learning. In *International MICCAI Brainlesion Workshop*, pages 456–463. Springer, 2018.

[55] Christen D Barras, Brian M Tress, Soren Christensen, Lachlan MacGregor, Marnie Collins, Patricia M Desmond, Brett E Skolnick, Stephan A Mayer, Joseph P Broderick, Michael N Diringer, et al. Density and shape as ct predictors of intracerebral hemorrhage growth. *Stroke*, 40(4):1325–1331, 2009.

[56] Negar Farzaneh, SM Reza Soroushmehr, Craig A Williamson, Cheng Jiang, Ashok Srinivasan, Jayapalli R Bapuraj, Kevin R Ward, Frederick K Korley, and Kayvan Najarian. Automated subdural hematoma segmentation for traumatic brain injured (tbi) patients. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3069–3072. IEEE, 2017.

[57] Douglas K Rex, C Richard Boland, Jason A Dominitz, Francis M Giardiello, David A Johnson, Tonya Kaltenbach, Theodore R Levin, David Lieberman, and Douglas J Robertson. Colorectal cancer screening: recommendations for physicians and patients from the us multi-society task force on colorectal cancer. *Gastroenterology*, 153(1):307–323, 2017.

[58] US Preventive Services Task Force et al. Screening for colorectal cancer: Us preventive services task force recommendation statement. *Annals of internal medicine*, 149(9):627, 2008.

[59] JJ Farrell and LS Friedman. The management of lower gastrointestinal bleeding. *Alimentary pharmacology & therapeutics*, 21(11):1281–1298, 2005.

[60] Vivek Chaudhry, Matthew J Hyser, Vicente H Gracias, and Frederick C Gau. Colonoscopy: The initial test for acute lower gastrointestinal bleeding/discussion. *The American Surgeon*, 64(8):723, 1998.

[61] Raj J Shah, Cecilia Fenoglio-Preiser, Brian L Bleau, and Ralph A Giannella. Usefulness of colonoscopy with biopsy in the evaluation of patients with chronic diarrhea. *The American journal of gastroenterology*, 96(4):1091–1095, 2001.

[62] David T Rubin, Ashwin N Ananthakrishnan, Corey A Siegel, Bryan G Sauer, and Millie D Long. Acg clinical guideline: ulcerative colitis in adults. *American Journal of Gastroenterology*, 114(3):384–413, 2019.

[63] L Peyrin-Biroulet, W Sandborn, BE Sands, W Reinisch, W Bemelman, RV Bryant, G d'Haens, I Dotan, M Dubinsky, B Feagan, et al. Selecting therapeutic targets in inflammatory bowel disease (stride): determining therapeutic goals for treat-to-target. *American Journal of Gastroenterology*, 110(9):1324–1338, 2015.

[64] Food, Drug Administration, et al. Ulcerative colitis: clinical trial endpoints guidance for industry. draft guidance. report no. 15028dft. Technical report, doc 07/27/16 UCM515143. Silver Spring, MD:: FDA, 2016: 19.

[65] L Peyrin-Biroulet, W Sandborn, BE Sands, W Reinisch, W Bemelman, RV Bryant, G D Haens, I Dotan, M Dubinsky, B Feagan, et al. O donnell s, pariente b, winer s, hanauer s, colombel jf (2015) selecting therapeutic targets in inflammatory bowel disease (stride): determining therapeutic goals for treat-to-target. *Am J Gastroenterol*, 110:1324–1328.

[66] Ulcerative Colitis. Clinical trial endpoints guidance for industry. *US Food and Drug Administration*, 2019.

[67] Ryan W Stidham, Wenshuo Liu, Shrinivas Bishu, Michael D Rice, Peter DR Higgins, Ji Zhu, Brahmajee K Nallamothu, and Akbar K Waljee. Performance of a deep learning model vs human reviewers in grading endoscopic disease severity of patients with ulcerative colitis. *JAMA network open*, 2(5):e193963–e193963, 2019.

[68] Kenneth W Schroeder, William J Tremaine, and Duane M Ilstrup. Coated oral 5-aminosalicylic acid therapy for mildly to moderately active ulcerative colitis. *New England Journal of Medicine*, 317(26):1625–1629, 1987.

[69] Jean Frédéric Colombel, Paul Rutgeerts, Walter Reinisch, Dirk Esser, Yanxin Wang, Yinghua Lang, Colleen W Marano, Richard Strauss, Björn J Oddens, Brian G Feagan, et al. Early mucosal healing with infliximab is associated with improved long-term clinical outcomes in ulcerative colitis. *Gastroenterology*, 141(4):1194–1201, 2011.

[70] Manuel Barreiro-de Acosta, Nicolau Vallejo, Daniel de la Iglesia, Laura Uribarri, Iria Bastón, Rocío Ferreiro-Iglesias, Aurelio Lorenzo, and J Enrique Domínguez-Muñoz. Evaluation of the risk of relapse in ulcerative colitis according to the degree of mucosal healing (mayo 0 vs 1): a longitudinal cohort study. *Journal of Crohn's and Colitis*, 10(1):13–19, 2016.

[71] Simon PL Travis, Dan Schnell, Piotr Krzeski, Maria T Abreu, Douglas G Altman, Jean-Frédéric Colombel, Brian G Feagan, Stephen B Hanauer, Marc Lémann, Gary R Lichtenstein, et al. Developing an instrument to assess the endoscopic severity of ulcerative colitis: the ulcerative colitis endoscopic index of severity (uceis). *Gut*, 61(4):535–542, 2012.

[72] Peter B Cotton and Christopher B Williams. *Practical gastrointestinal endoscopy: the fundamentals*. John Wiley & Sons, 2008.

[73] Kristoffer Derwinger and Bengt Gustavsson. Variations in demography and prognosis by colon cancer location. *Anticancer research*, 31(6):2347–2350, 2011.

[74] Seth M Steinberg, Jamie S Barkin, Richard S Kaplan, and Donald M Stablein. Prognostic indicators of colon tumors. the gastrointestinal tumor study group experience. *Cancer*, 57(9):1866–1870, 1986.

[75] Danish Abdul Aziz, Maryum Moin, Atif Majeed, Kamran Sadiq, and Abdul Gaffar Biloo. Paediatric inflammatory bowel disease: Clinical presentation and disease location. *Pakistan journal of medical sciences*, 33(4):793, 2017.

[76] Trung Duc Than, Gursel Alici, Hao Zhou, and Weihua Li. A review of localization systems for robotic endoscopic capsules. *IEEE transactions on biomedical engineering*, 59(9):2387–2399, 2012.

[77] Mohammad Ali Armin. *Automated visibility map from colonoscopy video to support clinical diagnosis and improve the quality of colonoscopy*. PhD thesis, University of Canberra, 2016.

[78] Jianfei Liu, Kalpathi R Subramanian, and Terry S Yoo. An optical flow approach to tracking colonoscopy video. *Computerized Medical Imaging and Graphics*, 37(3):207–223, 2013.

[79] Cristian Ballesteros, Maria Trujillo, Claudia Mazo, Deisy Chaves, and Jesus Hoyos. Automatic classification of non-informative frames in colonoscopy videos using texture analysis. In César Beltrán-Castañón, Ingela Nyström, and Fazel Famili, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 401–408, Cham, 2017. Springer International Publishing.

[80] ABM Islam, Ali Alammari, JungHwan Oh, Wallapak Tavanapong, Johnny Wong, and Piet C de Groen. Non-informative frame classification in colonoscopy videos using cnns. In *Proceedings of the 2018 3rd International Conference on Biomedical Imaging, Signal Processing*, pages 53–60. ACM, 2018.

[81] Mohammad Ali Armin, Girija Chetty, Fripp Jurgen, Hans De Visser, Cedric Dumas, Amir Fazlollahi, Florian Grimpen, and Olivier Salvado. Uninformative frame detection in colonoscopy through motion, edge and color features. In *International Workshop on Computer-Assisted and Robotic Endoscopy*, pages 153–162. Springer, 2015.

[82] Heming Yao, Ryan W Stidham, Reza Soroushmehr, Jonathan Gryak, and Kayvan Najarian. Automated detection of non-informative frames for colonoscopy through a combination of deep learning and feature extraction. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2402–2406. IEEE, 2019.

[83] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039, 2015.

[84] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5038–5047, 2017.

[85] Kaixuan Wang and Shaojie Shen. Flow-motion and depth network for monocular stereo and beyond. *IEEE Robotics and Automation Letters*, 5(2):3307–3314, 2020.

[86] Richard Szeliski. Prediction error as a quality metric for motion and stereo. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 781–788. IEEE, 1999.

[87] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017.

[88] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2022–2030, 2018.

[89] Frank Steinbrücker, Jürgen Sturm, and Daniel Cremers. Real-time visual odometry from dense rgb-d images. In *2011 IEEE international conference on computer vision workshops (ICCV Workshops)*, pages 719–722. IEEE, 2011.

[90] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 7286–7291. IEEE, 2018.

[91] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2018.

[92] Tianwei Shen, Zixin Luo, Lei Zhou, Hanyu Deng, Runze Zhang, Tian Fang, and Long Quan. Beyond photometric loss for self-supervised ego-motion estimation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6359–6365. IEEE, 2019.

[93] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018.

[94] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[95] V Madhu Babu, Kaushik Das, Anima Majumdar, and Swagat Kumar. Undemon: Unsupervised deep network for depth and ego-motion estimation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1082–1088. IEEE, 2018.

[96] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *Advances in neural information processing systems*, pages 35–45, 2019.

[97] Mehrdad Hosseinzadeh, Ramin Fahimi, Yang Wang, et al. Unsupervised learning of camera pose with compositional re-estimation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 11–20, 2020.

[98] Mohammad Ali Armin, Nick Barnes, Jose Alvarez, Hongdong Li, Florian Grimpen, and Olivier Salvado. Learning camera pose from optical colonoscopy frames through deep convolutional neural network (cnn). In *Computer assisted and robotic endoscopy and clinical image-based procedures*, pages 50–59. Springer, 2017.

[99] Mehmet Turan, Yasin Almalioglu, Helder Araujo, Ender Konukoglu, and Metin Sitti. Deep endovo: A recurrent convolutional neural network (rcnn) based visual odometry approach for endoscopic capsule robots. *Neurocomputing*, 275:1861–1870, 2018.

[100] Tsai Ping-Sing and Mubarak Shah. Shape from shading using linear approximation. *Image and Vision computing*, 12(8):487–498, 1994.

[101] Mehmet Turan, Evin Pinar Ornek, Nail Ibrahimli, Can Giracoglu, Yasin Almalioglu, Mehmet Fatih Yanik, and Metin Sitti. Unsupervised odometry and depth learning for endoscopic capsule robots. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1801–1807. IEEE, 2018.

[102] Daniel Freedman, Yochai Blau, Liran Katzir, Amit Aides, Ilan Shimshoni, Danny Veikherman, Tomer Golany, Ariel Gordon, Greg Corrado, Yossi Matias, et al. Detecting deficient coverage in colonoscopies. *arXiv preprint arXiv:2001.08589*, 2020.

[103] Ling Li, Xiaojian Li, Shanlin Yang, Shuai Ding, Alireza Jolfaei, and Xi Zheng. Unsupervised learning-based continuous depth and motion estimation with monocular endoscopy for virtual reality minimally invasive surgery. *IEEE Transactions on Industrial Informatics*, 2020.

[104] David Recasens, José Lamarca, José M Fácil, JMM Montiel, and Javier Civera. Endo-depth-and-motion: Localization and reconstruction in endoscopic videos using depth networks and photometric constraints. *arXiv preprint arXiv:2103.16525*, 2021.

[105] Kutsev Bengisu Ozyoruk, Guliz Irem Gokceler, Taylor L Bobrow, Gulfize Coskun, Kagan Incetan, Yasin Almalioglu, Faisal Mahmood, Eva Curto, Luis Perdigoto, Marina Oliveira, et al. Endoslam dataset and an unsupervised monocular visual odometry and depth estimation approach for endoscopic videos. *Medical image analysis*, 71:102058, 2021.

[106] Jason K Hou, Mimi Tan, Ryan W Stidham, John Colozzi, Devon Adams, Hashem El-Serag, and Akbar K Waljee. Accuracy of diagnostic codes for identifying patients with ulcerative colitis and crohn's disease in the veterans affairs health care system. *Digestive diseases and sciences*, 59(10):2406–2410, 2014.

[107] Bruce E Sands, William J Sandborn, Remo Panaccione, Christopher D O'Brien, Hongyan Zhang, Jewel Johanns, Omoniyi J Adedokun, Katherine Li, Laurent Peyrin-Biroulet, Gert Van Assche, et al. Ustekinumab as induction and maintenance therapy for ulcerative colitis. *New England Journal of Medicine*, 381(13):1201–1214, 2019.

[108] Geert D'haens, William J Sandborn, Brian G Feagan, Karel Geboes, Stephen B Hanauer, E Jan Irvine, Marc Lémann, Philippe Marteau, Paul Rutgeerts, Jurgen Schölmerich, et al. A review of activity indices and efficacy end points for clinical trials of medical therapy in adults with ulcerative colitis. *Gastroenterology*, 132(2):763–786, 2007.

[109] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[110] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[111] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5695–5701. IEEE, 2006.

[112] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.

[113] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016.

[114] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016.

[115] David Prokhorov, Dmitry Zhukov, Olga Barinova, Konushin Anton, and Anna Vorontsova. Measuring robustness of visual slam. In *2019 16th International Conference on Machine Vision Applications (MVA)*, pages 1–6. IEEE, 2019.

[116] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Computer Architecture Letters*, 13(04):376–380, 1991.

[117] Nariman Noorbakhsh-Sabet, Ramin Zand, Yanfei Zhang, and Vida Abedi. Artificial intelligence transforms the future of health care. *The American journal of medicine*, 132(7):795–801, 2019.

[118] S Caccomo. Fda permits marketing of artificial intelligence algorithm for aiding providers in detecting wrist fractures. *FDA News Release*, 2018.

[119] Monika A Myszczynska, Poojitha N Ojamies, Alix MB Lacoste, Daniel Neil, Amir Saffari, Richard Mead, Guillaume M Hautbergue, Joanna D Holbrook, and Laura Ferraiuolo. Applications of machine learning to diagnosis and treatment of neurodegenerative diseases. *Nature Reviews Neurology*, 16(8):440–456, 2020.

[120] Joeky T Senders, Patrick C Staples, Aditya V Karhade, Mark M Zaki, William B Gormley, Marike LD Broekman, Timothy R Smith, and Omar Arnaout. Machine learning and neurosurgical outcome prediction: a systematic review. *World neurosurgery*, 109:476–486, 2018.

[121] Heming Yao, Craig Williamson, Jonathan Gryak, and Kayvan Najarian. Automated hematoma segmentation and outcome prediction for patients with traumatic brain injury. *Artificial Intelligence in Medicine*, 107:101910, 2020.

[122] Thomas Davenport and Ravi Kalakota. The potential for artificial intelligence in healthcare. *Future healthcare journal*, 6(2):94, 2019.

[123] Lotfi A Zadeh. Fuzzy logic and approximate reasoning. *Synthese*, 30(3):407–428, 1975.

[124] Tomohiro Takagi and Michio Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE transactions on systems, man, and cybernetics*, (1):116–132, 1985.

[125] Kit Yan Chan, Sai-Ho Ling, Tharam Singh Dillon, and Hung T Nguyen. Diagnosis of hypoglycemic episodes using a neural network based rule discovery system. *Expert Systems with Applications*, 38(8):9799–9808, 2011.

[126] Ridong Zhang and Jili Tao. A nonlinear fuzzy neural network modeling approach using an improved genetic algorithm. *IEEE Transactions on Industrial Electronics*, 65(7):5882–5892, 2017.

[127] Grigory Mikhalkin. Tropical geometry and its applications. *arXiv preprint math/0601041*, 2006.

[128] Kishan S Parikh, Kavita Sharma, Mona Fiuzat, Howard K Surks, Jyothis T George, Narimon Honarpour, Christopher Depre, Patrice Desvigne-Nickens, Richard Nkulikiyinka, Gregory D Lewis, et al. Heart failure with preserved ejection fraction expert panel report: current controversies and implications for clinical trials. *JACC: Heart Failure*, 6(8):619–632, 2018.

[129] EJ Benjamin, SS Virani, CW Callaway, AM Chamberlain, AR Chang, S Cheng, SE Chiuve, M Cushman, FN Delling, R Deo, et al. American heart association council on e, prevention statistics c, stroke statistics s (2018) heart disease and stroke statistics-2018 update: a report from the american heart association. *Circulation*, 137(12):e67–e492.

[130] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[131] Brent Mittelstadt, Chris Russell, and Sandra Wachter. Explaining explanations in ai. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 279–288, 2019.

[132] Gilles Louppe. Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502*, 2014.

[133] Fred Hohman, Haekyu Park, Caleb Robinson, and Duen Horng Polo Chau. S ummit: Scaling deep learning interpretability by visualizing activation and attribution summarizations. *IEEE transactions on visualization and computer graphics*, 26(1):1096–1106, 2019.

[134] Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158, 2012.

[135] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1721–1730, 2015.

[136] A Stoica. Synaptic and somatic operators for fuzzy neurons: which t-norms to choose? In *Proceedings of North American Fuzzy Information Processing*, pages 55–58. IEEE, 1996.

[137] J-SR Jang. Anfis: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3):665–685, 1993.

[138] Ali Firat Cabalar, Abdulkadir Cevik, and Candan Gokceoglu. Some applications of adaptive neuro-fuzzy inference system (anfis) in geotechnical engineering. *Computers and Geotechnics*, 40:14–33, 2012.

[139] Majdi Al-Mahasneh, Mohannad Aljarrah, Taha Rababah, and Muhammad Alu'datt. Application of hybrid neural fuzzy system (anfis) in food processing and technology. *Food engineering reviews*, 8(3):351–366, 2016.

[140] Heming Yao, Keith D Aaronson, Lu Lu, Jonathan Gryak, Kayvan Najarian, and Jessica R Golbus. Using a fuzzy neural network in clinical decision support for patients with advanced heart failure. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 995–999. IEEE, 2019.

[141] Michio Sugeno and Kazuo Tanaka. Successive identification of a fuzzy model and its applications to prediction of a complex system. *Fuzzy sets and systems*, 42(3):315–334, 1991.

[142] Chuen-Tsai Sun. Rule-base structure identification in an adaptive-network-based fuzzy inference system. *IEEE Transactions on Fuzzy Systems*, 2(1):64–73, 1994.

[143] Mahardhika Pratama, Jie Lu, Edwin Lughofer, Guangquan Zhang, and Meng Joo Er. An incremental learning of concept drifts using evolving type-2 recurrent fuzzy neural networks. *IEEE Transactions on Fuzzy Systems*, 25(5):1175–1192, 2016.

[144] Omar Adil M Ali, Aous Y Ali, and Balasem Salem Sumait. Comparison between the effects of different types of membership functions on fuzzy logic controller performance. *International Journal*, 76:76–83, 2015.

[145] J Gayathri Monicka, N Guna Sekhar, and K Ramash Kumar. Performance evaluation of membership functions on fuzzy logic controlled ac voltage controller for speed control of induction motor drive. *International Journal of Computer Applications*, 13(5):8–12, 2011.

[146] Ali Sadollah. *Fuzzy Logic Based in Optimization Methods and Control Systems and Its Applications*. BoD–Books on Demand, 2018.

[147] Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*, 2019.

[148] Saroj K Meher. A new fuzzy supervised classification method based on aggregation operator. In *2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System*, pages 876–882. IEEE, 2007.

[149] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017.

[150] Ching-Hua Weng, Ying-Hsiu Lai, and Shang-Hong Lai. Driver drowsiness detection via a hierarchical temporal deep belief network. In *Asian Conference on Computer Vision*, pages 117–133. Springer, 2016.

[151] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. Openface: an open source facial behavior analysis toolkit. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–10. IEEE, 2016.

[152] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[153] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, pages 1135–1143, 2015.

[154] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[155] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[156] Chunming Li, Chenyang Xu, Changfeng Gui, and Martin D Fox. Distance regularized level set evolution and its application to image segmentation. *IEEE Transactions on Image Processing*, 19(12):3243–3254, 2010.

[157] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, Sabine Süsstrunk, et al. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[158] Olivier Rioul and Martin Vetterli. Wavelets and signal processing. *IEEE signal processing magazine*, 8(4):14–38, 1991.

[159] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[160] SG Machado, Gordon D Murray, and GM Teasdale. Evaluation of designs for clinical trials of neuroprotective agents in head injury. *Journal of neurotrauma*, 16(12):1131–1138, 1999.

[161] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[162] S Sadahiro, T Ohmura, Y Yamada, T Saito, and Y Taki. Analysis of length and surface area of each segment of the large intestine according to age, sex and physique. *Surgical and Radiologic Anatomy*, 14(3):251–257, 1992.

[163] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[164] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[165] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.