Supervised Deep Learning with Finite Element Synthetic Data for Force Estimation in Robotic-assisted Surgery

Kian Mirniazy

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Mechanical Engineering) at

Concordia University

Montréal, Québec, Canada

June 2022

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Kian Mirniazy**

Entitled: **Supervised Deep Learning with Finite Element Synthetic Data for Force Estimation in Robotic-assisted Surgery**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Mechanical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

______________________________ Chair
Dr. Ion Stiharu

______________________________ Examiner
Dr. Arash Mohammadi

______________________________ Examiner
Dr. Ion Stiharu

______________________________ Supervisor
Dr. Javad Dargahi

Approved by ______________________________
Martin D. Pugh, Chair
Department of Mechanical and Industrial Engineering

May 2022

____
Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Supervised Deep Learning with Finite Element Synthetic Data for Force Estimation in Robotic-assisted Surgery

Kian Mirniazy

The prevalence of robot-assisted minimally invasive surgery on the liver has increased exponentially. Having accurate, real-time knowledge of force during robotic-assisted surgical procedures is vital for safe surgery. Many techniques have been proposed in the literature to tackle this concern, from deploying force sensors to physics-based modeling of the robot and, more recently, learning-based force prediction. For a high-fidelity force measurement, sensors should be integrated at the instrument's tip, close to the surgical site, which brings sterilization, biocompatibility, and MRI compatibility concerns. On the other hand, Dynamic robot modeling may be precise in a specific setting, but it suffers from the lack of generalization encountering unseen settings. Considering the drawbacks and deficits of mentioned methods, indirect force estimation via deflection measurement through imaging techniques is investigated as an alternative solution, generally done via machine learning methods. Almost all previous studies are either supervised learning, where data are labeled with ex-vivo ground truth, or unsupervised or semi-supervised learning, where outcomes are promising but not adequate. This study investigated indirect force prediction for the human liver through a developed deep autoencoder model as a supervised deep learning method trained via synthetic data generated by finite element (FE) simulation. This method took advantage of various patient-specific livers parameters and geometries extracted from CT images. The Hyperelastic modeling of the soft tissue is considered and assessed with various hyperelastic models. The uncertainty due to the surgical tool's occlusion is addressed in this model, and a novel state vector was proposed to improve the accuracy and generalisability of the prediction. In addition, the impact of the bounded region on the model's accuracy was evaluated. It was shown that the proposed method could predict the external force on an unseen tissue with different geometry and mechanical properties. The accuracy of force prediction considering tool occlusion noise diminishes by 4.2 percent, which is in an acceptable range. The accuracy of presented model for various scenarios ranges from 95 to 88 percent. Model's results have been evaluated by predicting the force encountering the surface deformation of an unseen liver geometry and constitutive model where the mean absolute error of prediction is 0.249 Newton.

Acknowledgments

Throughout the writing of this thesis, I have received a remarkable deal of support and assistance. First, I must express my massive gratitude to my supervisor, Professor Javad Dargahi, whose continued support, and guidance enlightened the path. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

I would like to acknowledge my colleagues and friends Amir Molaei and Nima Amoei for their wonderful collaboration, empathy and support.

I want to thank my best friend, Moe karamzadeh with his persistent help. Who motivated me through facing and overcoming the challenges and barriers. In addition, I would like to thank my wife, Yalda, for her selfless support and companionship. This would not be possible without her. Lastly, I am more than grateful for having my mother whose unlimited energy and support were always felt wholeheartedly.

Finally, many thanks to Concordia University and its staff for the great facilities, events, and classes which helped me a lot on this path.

Contents

List of Figures

List of Tables

Nomenclature

$\bar{I}_i$	Second deviatoric strain invariant
λ	Stretch ratio
μ	Lame constant
b	bias term
C_i	Mooney Rivlin Parameters
C_{def}	ROI deformed centroid
C_{undef}	Undeformed liver model centroid
E	Green-Lagrangian strain tensor
F_i	Force component in direction i
I_{xyx}	Moment of Inertia
P	External force exertion coordinate
S	State Vector
T_i	Torque component in direction i
W	Strain energy density function
W^L	Weight link at layer L
ϵ_i	Engineering strain

σ^{PK2}	2nd Piola-Kirchhoff stress
σ_i	Cauchy stress
T	Principal component matrix
W	Eigen vector of covariance Matrix

Chapter 1

Introduction

1.1 Background and Motivation

Sensing Haptic feedback has been undoubtedly an essential enhancement for minimally invasive surgeries, especially in robotic aided surgeries where tangible force feedback plays a vital role in maintaining the efficiency of the procedure, from increasing the surgeon's proficiency to manually palpating tissue, assessing tissue characteristics to prevent applying excessive forces which leads to a safer surgery that decreases trauma and damage to healthy tissue and also prevents the exertion of insufficient force that might be applied when grasping tissue or sutures, leading to slippage, loss of control, and loose intracorporeal knots.

Safe and effective tissue handling is an important skill, and sensing haptic feedback plays a vital role in maintaining the efficiency of the surgical procedure. To date, several research have been devised to tackle this problem. One method to acquire haptic feedback can be provided by mounted force sensors in the environment, such as external force sensors or at the end-effector, i.e., internal sensors. Even though commercially available force sensors have been proven to be effective for measuring forces in various applications, the particular case of surgical environment imposes severe constraints on their applicability and efficacy in terms of biocompatibility and sterilizability [1]. Furthermore, the surgical instrument is guided to the surgical site through a trocar. The interaction forces in the surgical site are dissipated due to frictional forces at the instrument's entry port. Additionally, adding force sensing capabilities for those robotic systems primarily designed without

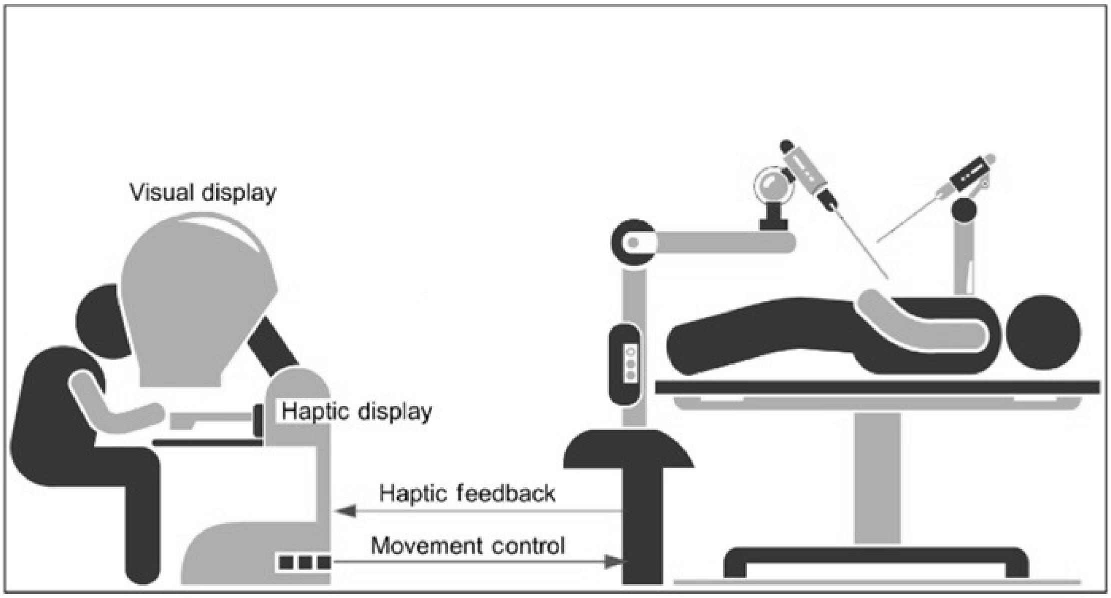


Figure 1.1: Haptic feedback enhances surgeon's dexterity and improves post operation outcomes

force sensors is not easily affordable. In addition, force sensors are mostly non-functional close to a magnetic field, such as MRI in the surgical site [2, 3].

In the context of tissue handling and force modulation, different approaches are introduced to the surrogate measuring of tool-to-tissue interaction in robotic-assisted minimally invasive surgeries (RAMIS) [4].

Another approach is skill assessment, which can be done through video review using qualitative rubrics. Deficits in rating consistency, timeliness of feedback, and time demands on raters are the main drawbacks of this method. Obviously, this method does not provide real-time applicability in the clinical context. Robot dynamic modeling is another method that has been developed recently to estimate force at the tip by estimating the joint's torques. Due to the dynamics of the cable-driven arm, these models typically do not generalize properly.

Learning-based force perception (LBFP) is introduced as a promising solution to overcome the limitations mentioned above. This method can use visual data acquired from the surgical site. In LBFP, deformation of the soft tissue can serve as a basis for the approximated force. In chapter 2, each method will be explained in details.

1.2 Research Objective and Aim

This study aims to present a learning-based model that deploys synthetic data of tissue deformation generated by finite element simulation of the patient-specific liver 3D model. The data mentioned above is labeled with relative external force and preoperative information to surrogate the utilization of a force sensor to measure forces where it is not feasible to employ an actual sensor. Deep neural network (DNN) as one of the most advanced techniques for nonlinear regression problems was opted. The complex modeling capabilities of a DNN, combined with its high accessibility nowadays, provide new opportunities in the field of Surgical innovations.

It is worth mentioning that most previous LBFP studies employed labeled and acquired data experimentally. Typically the experiment is operated in an artificial phantom or a portion of an inanimate organ. Hence these data acquisitions are constrained to a limited number of samples, and the model shows a lower accuracy in encountering unseen biomechanical properties. One alternative to providing a training dataset is using a digital twin in a virtual environment. A digital twin is an ingenious concept that helps on organizing different areas of expertise, and a growing interest is observable in the engineering fields [5]. One important category of digital twins is the finite element model (FEM), which is used to find the system's behavior under external loads. Examples of using FE generated data for training datasets can be found in [6, 7, 8], with structural health monitoring application

The current master's thesis objective is to develop a framework that deploys DNN for real-time force prediction concerning surgical invivo conditions such as boundary conditions, organ hyperelasticity, and 3D geometry. The model is trained on a synthetic dataset generated by FEM-based digital twins of liver tissue in MIS to provide a novel approach to estimate force based on visual information of tissue in minimally invasive surgeries. The goal within the time span of this thesis is to enhance the accuracy and generalisability of learning-based force estimation DNN to predict the force exerted on the liver surface based on the bounded region of MIS and preoperative information of the organ.

1.3 Thesis Contribution

- 3D reconstruction of patient-specific liver models associated with its hyperelastic constitutive model and boundary interaction.
- Finite element modeling of soft liver tissue with ANSYS FEA package.
- Parameterize and automatize the FEM pre and post-processing steps to retrieve over 23000 simulations for various design points (force exertion) and 3D liver geometries.
- Propose and implement a deformation-based Deep learning model to take ROI deformation and map it to force components.
- Propose a novel state vector that comprises pre-deformation state geometrical and constitutive features of the organ for a more accurate deep neural network.

1.4 Thesis Layout

This thesis continues with a comprehensive literature review in chapter 2, where previous studies and background knowledge and relevant approaches and methods are widely discussed. In addition, it covers recent advances in machine and deep learning applications in force feedback estimation. It concludes with a reflection on the found information and the substantiation of the research gap this work fits.

To overcome the drawbacks mentioned in chapter 2, The generated dataset based on the patient's CT scan is explained, and the finite element modeling and related considerations such as hyperelastic modeling, boundary conditions, and parametrization of the simulation to grasp sufficient amount of data is discussed in chapter 3 in details.

In chapter 4, Machine learning and Deep learning implementation have been elaborated, and the proposed method and its features are explained. The mathematical concept beyond the method would be pointed out. Chapter 5 includes result that is obtained from our proposed model and the comparison with previous studies. Finally, chapter 6 concludes and answers devised questions and objectives of this study.

Chapter 2

Literature Review

2.1 Force and Haptic Feedback

Haptic perception is the information and control related to or proceeding from the sense of touch that can be kinaesthetic (the force and position in the joints and muscles) and cutaneous (tactile, related to the skin) [9]. Haptic feedback has the potential to provide superior performance and reliability in master-slave robotic applications [10]. For teleoperation, measuring forces that cause deformation and translation will provide the required feedback[11].

The most common solution to measure forces in experimental systems is to mount regular, commercially available force/torque sensors to the robot externally [12, 13, 14]. MC2E robot is an example that can measure the distal organ-instrument interactions with a sensor mounted outside of a patient's region [15]. They introduced a compact and lightweight robot for MIS mounted on the patient and offered a specialized kinematic design dedicated to MIS, which provides an invariant center at the fulcrum point. Another project was the BlueDRAGON which was developed at the University of Washington that records both the kinematics and dynamics (positions and forces/torques in 6 DOF) of two laparoscopic tools [16]. This technique can be practical for a range of applications, where the rigidity of the tools allows the accurate reading of the forces even further from the tip of the instrument. However, the main drawback is the dimensional constraints of surgical tools, especially during MIS procedures, where the oversized sensors can only be mounted relatively far from the tool tip, which results in less accuracy of the measurements due to high friction at the trocars [17].

One other general force measurement approach is integrating small size sensors that can be added to the instrument's tip within the surgical site. Many materials can be used where torques and forces cause an elastic deformation of the sensor bodies. Strain gauges are widely accepted. However, an array of them is needed to provide accurate force sensing in multiple degrees of freedom of motion. Several micro force sensors have been developed based on MEMS, Piezoelectric, optical, and Bragg sensors [18]. However, such technology has yet to see commercial application since its limitation to meet the criteria of biocompatibility, sterilizability, and cost-effectiveness. In addition, force sensors are mostly non-functional close to a magnetic field, such as MRI-guided surgeries [2, 3, 19].

To date, many approaches have been introduced to measure tool-to-tissue interaction in robotic-assisted minimally invasive surgeries [4]. From skill assessment to robot's dynamic models that estimate force at the tip by estimation of joint torques but due to dynamics of the cable-driven arm, these models typically do not generalize adequately.

2.1.1 Skill Assessment and Automated Performance Metrics

An alternative approach to improve surgeons' tissue handling and force modulation abilities is skill assessment that can be done through video review using qualitative rubrics [20, 21]. Deficits in rating consistency, timeliness of feedback, and time demands on raters are the main drawbacks of this method. Obviously, in the context of clinical application, this method can not provide real-time feedback. Another approach to evaluation is automated performance metrics (APMs) that are quantitative measures of skill [22]. While APMs for skill categories such as economy of motion and bi-manual dexterity are well known and easily measurable by querying the robot state and kinematics, APMs for measuring tissue handling are difficult to implement without either environmental or end-effector force sensing, leading to force APMs being excluded from clinical studies.

2.1.2 Robot's Dynamic Model

Robot's dynamic modeling is another method that has been recently deployed to estimate force at the end effector [23, 24, 25]. A sensorless model-based approach for contact force and torque

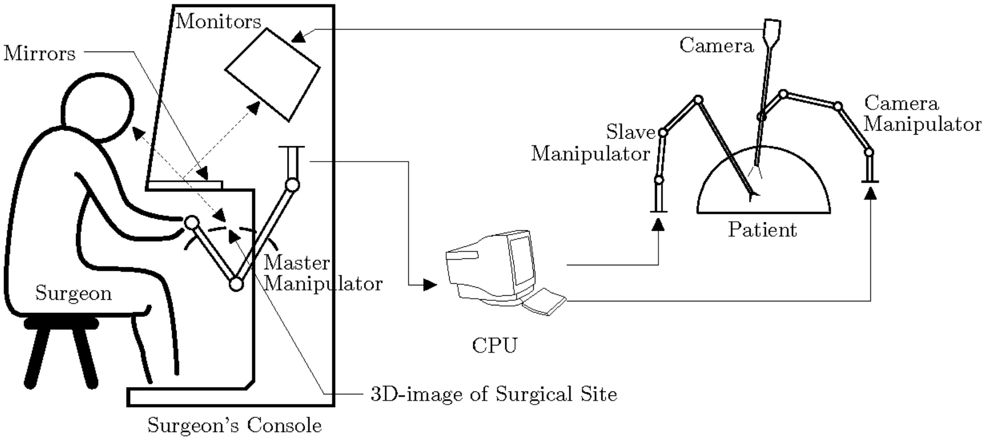


Figure 2.1: Robot-assisted minimally invasive surgery manipulation.

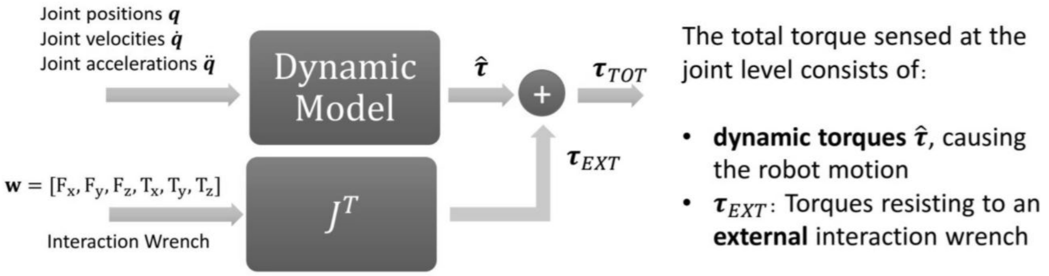

Joint positions $\boldsymbol{q}$
Joint velocities $\dot{\boldsymbol{q}}$
Joint accelerations $\ddot{\boldsymbol{q}}$
Dynamic Model
$\hat{\boldsymbol{\tau}}$
+
$\boldsymbol{\tau}_{TOT}$
$\boldsymbol{\tau}_{EXT}$
$\mathbf{w} = [F_x, F_y, F_z, T_x, T_y, T_z]$
Interaction Wrench
J^T
The total torque sensed at the joint level consists of:
• **dynamic torques** $\hat{\boldsymbol{\tau}}$, causing the robot motion
• $\boldsymbol{\tau}_{EXT}$: Torques resisting to an **external** interaction wrench

Figure 2.2: Dynamic model of dVRK slave arm to estimate the wrench interaction via joint torque

estimation. Pique et al. have modeled the dynamics of the dVRK slave arm to use the joint torques obtained from the measured motor currents and subtract the torques resulting from the dynamics of the robot arm, Fig. 2.2. Hence the residual torques are due to the external forces and torques acting on the tool, obtained through the inverse transpose of the Jacobian matrix. It is evident that due to the dynamics of the cable-driven arm, these models typically do not generalize adequately.

2.2 Deformation-based Force Perception

Haptic feedback can be derived from other sources of information, where the bending of the tools or an appropriate dynamic model of the soft tissue elasticity parameters can serve as a basis for approximate force computation [26]. Hence to overcome the mentioned limitation of other models and techniques, deformation-based force perception (DBFP) is introduced as a promising solution that uses the visual information of deformed tissue to compute the exerted force.

The history of combining force and visual feedback goes back to the late 90s when Zhou et al. presented a sensor fusion framework to integrate an optical microscope with a visually-servo micro-assembly system [27]. They measured the force in a force-controlled micro-positioning task based on the principles of optical beam deflection. Since then, many researchers have presented ideas to improve force measurement and estimation using visual feedback. This method has the advantage of using data acquired from imaging modalities along with additional references such as Stereo endoscopic images.

2.2.1 Image Processing and 3D Reconstruction for Visual Feedback

In order to estimate the interaction forces applied to a deformable object based on visual information, a broad range of image processing techniques can be utilized to collect accurate and reliable information. The vision system is required to detect the deformable objects in the captured images and track them over time [28]. To this end, image preprocessing, segmentation, and object recognition are carried out to provide visual feedback. The speed of the vision systems highly depends on the speed of the localization and recognition stages [29].

The image preprocessing includes image filtering, data extraction, and an optional data reduction step if the data is significantly more than needed for time-efficient visual information extraction. In image filtering, some of the low-level details of the image may be missed. Although preprocessing results in missing a portion of information, denoising helps the user extract more reliable information from the position of the deformable object. Considering the volume of visual data collected in the vision system, processing all of them in the subsequent steps increases the computation time excessively, which is a critical problem, especially in real-time applications.

Segmentation culminates in more time-efficient information extraction since it avoids the irrelevant computation of the whole environment. Segmentation helps the user to limit the tracking area to the close vicinity of the object, called the region of interest (ROI). Threshold, Edge detection, and region extraction are the best to known techniques for this purpose. The segmented image is then fed to the object recognition stage in which features of the object are extracted to be used in the following matching steps during a task. Feature detection is based on exploring a point of interest

(POI) in the image.

2.2.2 Physic-based Models

After the idea of vision and force sensing integration was presented [27], researchers developed and employed it, particularly for different tasks, including micromanipulation. Later, most of the research concentrated on image-based force measurement. The estimation has been taken towards techniques based on the model of the deformable objects. Most studies spotted problems with medical interventions in either exvivo or invivo scenarios. This method has the advantage of using data acquired from imaging modalities and additional references such as stereo endoscopic images. For instance, a vision-based contact force estimation method utilizing the FE model of the deformable object and its nodal displacements was presented in [30].

Another study measured the forces applied to the deformable tissues using visual feedback in a small single-port neurosurgical robot. Using a series of depth maps extracted from a stereo endoscope, they estimated the force. The reference shape of the soft tissue was obtained by processing the pre-operative CT images. Then, a surface mesh-based model making use of a spring-damper system was employed to model the object deformation [31].

The forces applied to soft tissue in a robotic surgery were estimated by Noohi et al. using a phenomenological nonlinear model and template matching-based methods for the deformation reconstruction [32]. They validated the estimation accuracy in an ex vivo experiment on a lamb liver. The advantage of this research was employing the concept of virtual template in the modeling of surface deformation without prior knowledge of the shape of the undeformed object.

In another study, the forces applied to a soft tissue were estimated using a force-displacement model created based on a second-order polynomial fitting. Considering tool-occluded images, the displacement at the tool-tissue contact point was determined by probabilistic tracking of the unrestricted movement of the points close to the tool's tip. The resulting 3D displacement was used to model 3D tissue deformations [33].

A method presented by Haouchine et al. constructs a depth map of the environment, which is mapped to a finite element mesh with properties consistent with the manipulated tissue [34]. They aim to recover the external force f from the displacement of the tip of the instrument. Therefore,

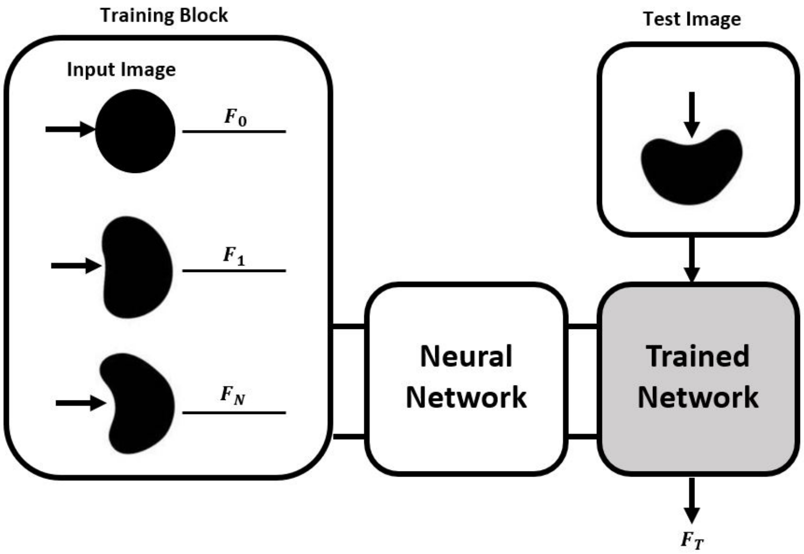


Figure 2.3: The first learning-based scheme to image-based force measurement [35]

they considered the 3D positions of the instrument as constraints that will impose the motion of the mechanical model. The displacement of the instrument will generate forces according to the stress-strain relation of the equation of deformation. Assuming the 3D displacement of the instrument tip is stored in the vector p, these constraints are modeled by adjoined 2m Lagrangian multipliers collected in vector l that yields the multiplier-augmented form. In an exvivo experiment, this method correctly approximated the trends of the interaction forces, but with an offset and scaling. The authors noted that the method's accuracy is dependent on the proposed anchoring points of the finite-element mesh.

2.2.3 Learning-based Approach

Attentions are towards using ML and DL to surrogate previous methods to measure interaction forces during the surgical procedure. Learning-based force estimation was represented for the first time in 2003 where the author proposed the application of artificial neural networks (ANN) [35]. They measured the forces applied to a deformable object based on template matching-based image processing algorithms in macro-scale and micro-scale scenarios as demonstrated in Fig. 2.3.

In general, the training of neural networks as an optimization problem requires a considerably large dataset, which is not easily accessible in the context of medical applications. Providing an appropriate training dataset, the time and computation strength needed for the training are contributing factors that should be considered, especially in the medical and healthcare systems. The successful application of the convolutional neural networks (CNN) in computer vision-based tasks such as pattern recognition enticed its utilization in medical applications.

2.2.4 Recurrent Neural Networks for Force Prediction

A recurrent neural network (RNN) is a type of artificial neural network which uses sequential data or time-series data. These deep learning algorithms are commonly used for ordinal or temporal problems. The backpropagation algorithm of a recurrent neural network is modified to include the unfolding in time to train the network's weights. This algorithm is based on computing the gradient vector and is called backpropagation in time or the BPTT algorithm. In force prediction of soft tissues, many researchers considered the previous state of tissue accountable for the final formation and dynamically analyzed it. RNN is the leading architecture that is used in these studies. Furthermore, It is occasionally seen that to prevent an expected deficit in RNN, vanishing gradient problem, long short term memory (LSTM) has been implemented. On the other hand, many consider a quasi-static process where initial (neutral) and final formation suffices to study tissue behavior adequately. Aviles and his fellow researchers initiated a series of research on the vision-based force prediction utilizing learning-based methods, a significant improvement in the LBFP. In the first step, they used an RNN, trained based on the Levenberg-Marquardt algorithm for a 3D vision-based force estimation task in MIS [36]. As mentioned above, RNNs are highly functional. There are complex dependencies between input and output variables of the network, input signals are significantly noisy, and considering temporal information is essential. The methodology comprises two modules of stereo vision and force measurement. They retrieved the deformation produced by the applied force using an energy minimization strategy and a cubic B-spline model of the object. In contrast, the latter extracted the relationship between the vision module's output and the robot's kinematic variables to calculate the force vector. The accuracy and computational performance of the method were validated by a static simulated tissue (heart) made of rubber silicone. The advantage of this

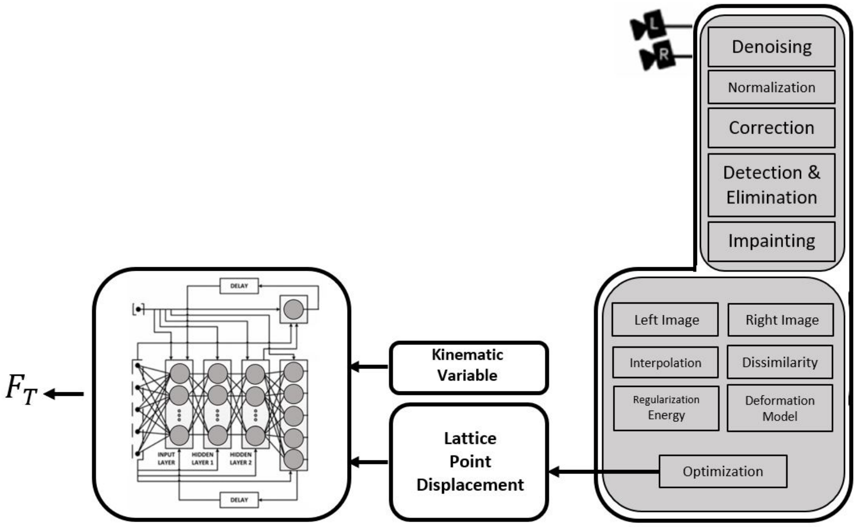


Figure 2.4: RNN-based scheme for vision based force prediction by [36]

method was that no knowledge of material properties was needed. The article did not provide detailed information about the network training and testing dataset. The model has been trained on a dataset with an ex-vivo setting.

Despite the robustness of this method with the ability to deal with noisy data and create complex input-output relations, the main concern with using RNNs for force estimation is the vanishing gradient problem [37]. The problem arises when the gradient of the error function is relevantly tiny, which prevents the network weights from being updated. The problem was tackled by employing a long short-term memory RNN architecture (RNN-LSTM) to preserve information for a more extended period, enforce constant error flow, and estimate the applied forces. This paper showed a performance improvement of up to 4% compared to the base research described.

In 2016 this research was further extended by improving the robustness and accuracy of the force estimation by integrating a fuzzy approach to the force estimation module, resulting in a deep neural-fuzzy system 2.5. This paper used the same approach, RNN-LSTM, to learn the properties of soft tissues [38]. The authors investigated the displacement error of soft tissue at the contact point in different force estimation architectures with and without neural networks or fuzzy systems and showed a significant improvement. They also compared the accuracy of the proposed approach with previously published methods and showed an improvement ranging from 35% to 85% [32, 39, 40].

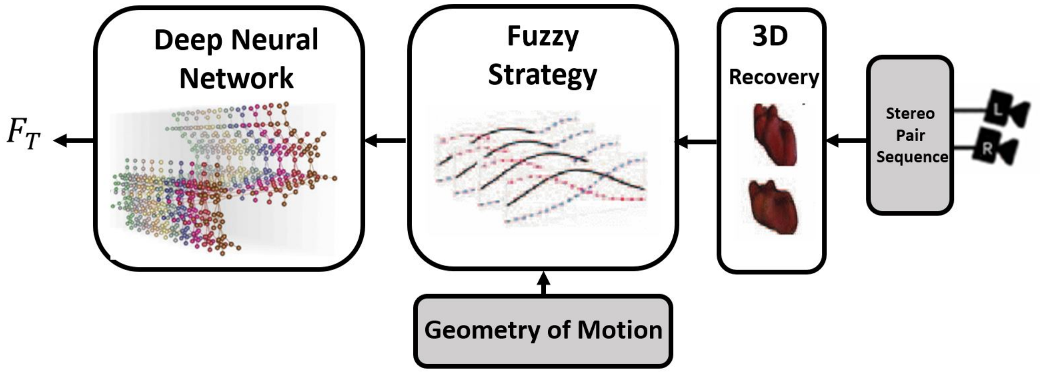


Figure 2.5: Deep neuro-fuzzy approaches to vision-based force estimation presented in [38]

They modified the vision module to accommodate complex behaviors of the deformation structure. They reformulated the total energy of the lattice points of the deformable tissue to tackle the tool partial-occlusion issue [41]. By dimensionality reduction via correlation-based feature selection and keeping only force related features, they reduced complexity and accelerated the performance of the deep network. The results showed the proposed method's robustness, accuracy, and stability over long periods. One limitation of the work was that the authors had not considered the organ dynamics during the experimental acquisition of the ex vivo datasets.

2.2.5 Convolutional Neural Network Force Prediction Models

As discussed in section 1, convolutional neural networks (CNN) have been widely used in literature to estimate force, based on a stereoscopic image of deformed tissue that is mainly labeled with the force, which is generally measured through an exvivo setting [42]. One of the main drawbacks of image-based CNN is the perturbation caused by the change in viewpoint and lighting. In literature, one of the most famous architectures that have been widely used for image-based classification and regression is a deep residual network and, specifically, ResNet-50 [43]. This model was the winner of the ImageNet challenge in 2015. The ResNet-50 model consists of 5 stages, each with a convolution and identity block. Each convolution block has three convolution layers, and each identity block also has three convolution layers. The ResNet-50 has over 23 million trainable parameters, as shown in Fig. 2.6 and a pre-trained version of the network trained on more than millions of images

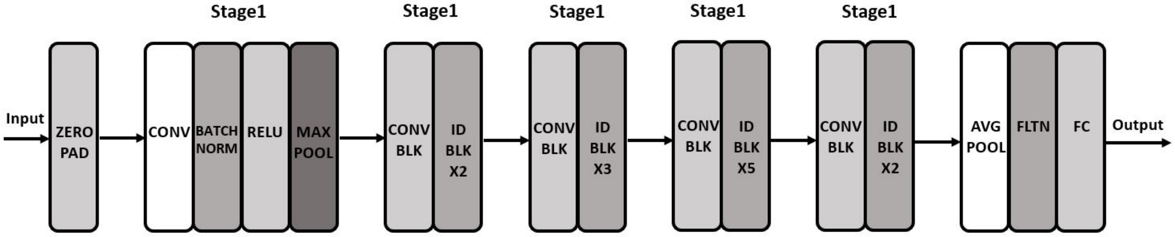


Figure 2.6: ResNet50, Deep residual network

executed on the ImageNet database. In literature, this architecture is the backbone of many models that deploy images to execute the task.

As it is pictured in Fig. 2.6 the Resnet_50 architecture contains the following element:

- A convolution with a kernel size of 7 * 7 and 64 different kernels all with a stride of size 2, giving us one layer.
- Max pooling with a stride size of 2.
- A 1 * 1 * 64 kernel following a 3 * 3 * 64 kernel and a 1 * 1 * 256 kernel, These three layers are repeated three times, giving us nine layers in this step.
- Next, see the kernel of 1 * 1 * 128 after that, a kernel of 3 * 3 * 128, and at last, a kernel of 1 * 1 * 512. This step was repeated four times so, giving us 12 layers in this step.
- After that, there is a kernel of 1 * 1 * 256 and two more kernels with 3 * 3 * 256 and 1 * 1 * 1024, and this is repeated 6 times, giving us a total of 18 layers.
- And then again a 1 * 1 * 512 kernel with two more of 3 * 3 * 512 and 1 * 1 * 2048 and this were repeated three times, giving us a total of 9 layers.
- After that, an average pool is done and end it with a fully connected layer containing 1000 nodes and a softmax function.

Gassert et al. estimated the interaction forces of instruments and tissue in a RAMIS directly from OCT image volumes [44]. They introduced a novel Siamese 3D CNN architecture that takes

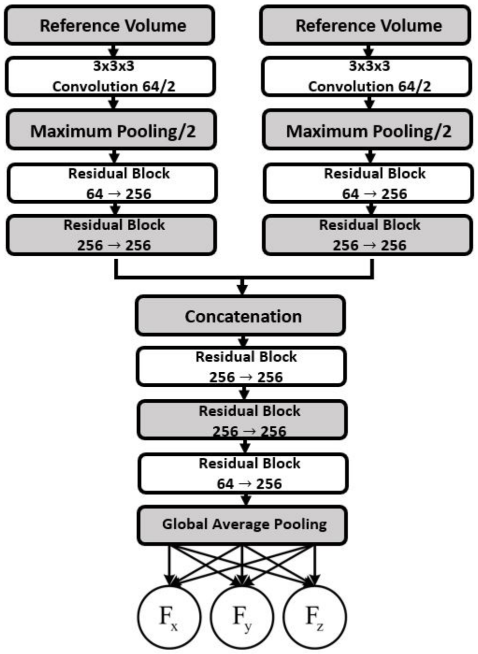


Figure 2.7: CNN based SIAMSE architecture presented by [44]

an undeformed reference volume and a deformed sample volume as inputs and outputs the three components of the force vector as shown in Fig. 2.7. The authors employed a deep residual architecture with bottlenecks to increase efficiency. They compared the Siamese approach to methods that use different volumes and 2D projections. A robotic setup was used to obtain ground-truth force vector for a silicone-made phantom and a porcine tissue.

The results showed that the accuracy of this method outperforms single-path methods. Moreover, the use of volume data resulted in significantly higher performance compared to processing only surface information.

Stereo images can be used in DBFP to construct a depth map of the environment, which is mapped to a finite element mesh with properties consistent with the manipulated tissue. Having a suitable biomechanical model of the soft tissue, the imaging techniques can be used to measure the deflection and find the interaction forces in real-time. Convolutional neural network encodes visual information. Apart from vision perception, like a surgeon's intuition in hand-eye coordination, some studies considered the robot state a condition vector comprising joint currents and the tool's tip position. In 2020, Chua et al. presented a force estimation neural network that uses RGB images and robot state as inputs [45]. Using a self-collected exvivo dataset, they compared the network to variants that included only a single input type. They evaluated how they generalized to new viewpoints, workspace positions, materials, and tools. They found that vision-based networks were sensitive to shifts in viewpoints, while state-only networks were robust to changes in the workspace. The network with both state and vision inputs had the highest accuracy for an unseen tool and was moderately robust to changes in viewpoints. They introduced a single-framed vision-state convolutional network that deploys a robot force-torque state and the vision acquired from monocular images on an artificial tissue.

$$S_{kin} = [P_i, O_i, Q_i] \tag{1}$$

They achieved a promising result that the network with robot state concatenated to CNN decreases force estimation error and generalizes better for unseen settings. However, in-vivo applicability remained questionable, and the model is trained based on a force sensor mounted in an experimental setting, so the moment is not considered actively.

2.2.6 Unsupervised Learning Models

In general, The models discussed in sections 2.2.4 and 2.2.5 are ANN-based methods that use supervised learning architectures to find the relation between vision features and the corresponding forces, which inevitably necessitate the utilization of ground-truth data. However, most of the video sequences related to robotic surgery are not provided with ground truth which is the main limitation.

In order to tackle this problem, researchers started to gradually transit from the supervised networks to unsupervised architectures. In 2018, Marban and fellow scientists used a semi-supervised learning framework to solve this issue. Hence, an unsupervised network, called convolutional auto-encoder (CAE), was employed to extract the feature vectors of the frames in an unlabeled video sequence [46]. Then, an LSTM network was trained in a supervised learning setting using available ground truth force data. The encoder network of the CAE, which was serially connected with the LSTM network, was trained partially to minimize the difference between the ground truth and the estimated force data. Fig. 2.8 and 2.9.

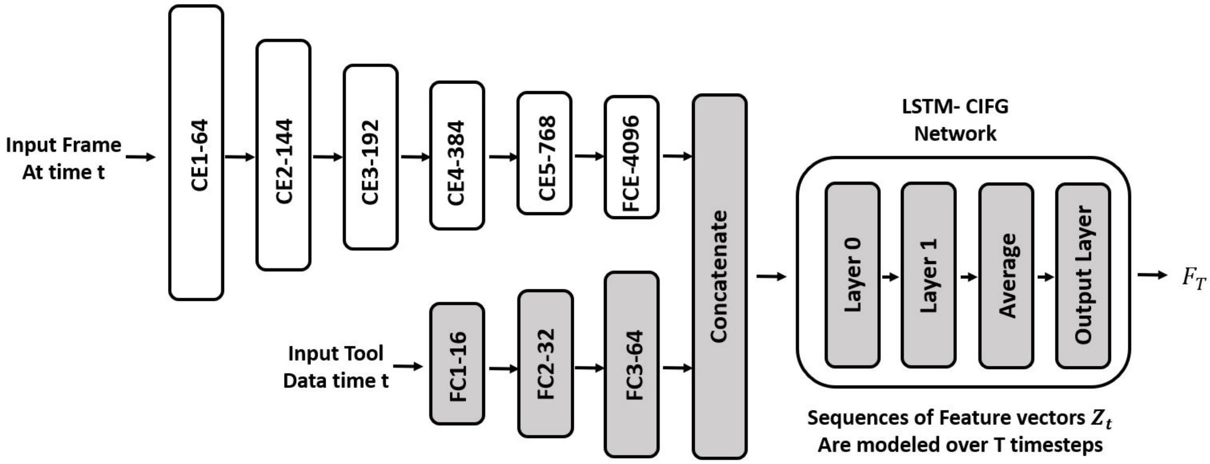


Figure 2.8: Encoder- LSTM model used in the semi-supervised learning stage in [46]

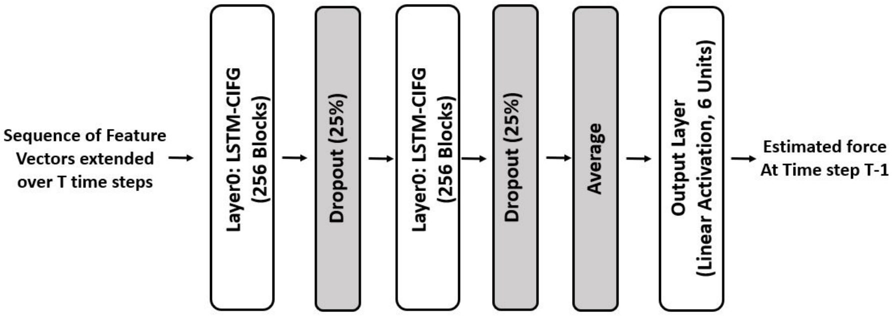

Sequence of Feature Vectors extended over T time steps
Layer0: LSTM-CIFG (256 Blocks)
Dropout (25%)
Layer0: LSTM-CIFG (256 Blocks)
Dropout (25%)
Average
Output Layer (Linear Activation, 6 Units)
Estimated force At Time step T-1

Figure 2.9: Design of the two-layer LSTM network with Coupled Input-Forget Gates (LSTM-CIFG) in [47]

Considering the scarcity of datasets addressing the force estimation tasks, the experiments were validated using a custom dataset experimentally collected by the authors. The proposed approach was experimentally validated on a silicone-made digestive apparatus. The results were promising, but the force estimation was not accurate enough for operational purposes. In 2019, Marban et al. presented a method for estimating a 6D vector of forces and torques (in the 3D space) in both pushing and pulling telesurgical tasks based on the CNN and LSTM networks [47].

Results obtained from these studies are remarkable, but the way they translate to RMIS is vague since they achieve high accuracy on relatively simple material compression tasks or use fewer complex manipulators and maneuvers. For instance, Unseen viewpoints and tissue biomechanical parameters can cause major error [48].

2.3 Research Gap

A thorough look over the aforementioned studies highlights that these models are either associated with the experimental dataset, i.e., exvivo data as labeled ground truth or designate an unsupervised approach where results are promising but not adequate in the clinical application context. Therefore, one of the main concerns in the presented solutions is the lack or absence of invivo considerations. Eventually, these models need to enhance the MI surgeons' perception of force magnitude where the organ is in the body with interaction with other organs and muscles, etc. In addition, in MIS, generally, a portion of the organ is observable through the stereoscopic camera, and assessment should be based on information conveyed from this segment. At the same time, the deformation occurs over the entire organ.

Another substantial drawback in previous studies is the lack of generalization. As most of the data are labeled and acquired experimentally, typically, the experiment is operated in an artificial phantom or a portion of an inanimate organ. Hence, these data acquisitions are constrained to a limited number of samples. The model shows a lower accuracy in encountering unseen material biomechanical properties considerations are missing in most of these models. On top of these limitations, in an ex-vivo setting, a simplified geometry is generally employed and geometrical. Boundary conditions do not address the actual status, diminishing the models' invivo compatibility.

2.4 Study Overview

One alternative to providing a training dataset in the absence of an adequate labeled dataset is a digital twin in a virtual environment. A digital twin is an ingenious concept that helps organize different areas of expertise, and a growing interest is observable in the engineering field. One important category of digital twins is the finite element model (FEM), which is used to find the system's behavior under external loads. Examples of using FE generated data for training dataset can be found in literature, as mentioned in chapter 1, with a structural health monitoring application.

In this study, finite element modeling (FEM) is used to provide synthetic data of liver deformation in external force where liver 3D geometries are extracted from CT scan images. In order to consider invivo compatibility, the dataset is presented by simulating patient-specific liver models with regard to the organ biomechanics and neighboring boundary conditions. Biomechanical characteristics of the soft tissue were obtained from experimental uniaxial, bi-axial, and shear stress tests from previous studies [49]. Developed models are used under various loading conditions, and the point cloud of the deformed shape is used as the primary training dataset. Note that, in the actual surgery, the point cloud of the deformed tissue can be obtained using imaging methods. Hence the proposed model is potentially a vision-based force prediction model. Considering the limitation of the field of view in the surgical site, the point cloud dataset is preprocessed to capture a limited region, referred to as the region of interest (ROI). To have a large enough dataset, over twenty-three thousand simulations have been retrieved on ANSYS software [50]. This process has been parameterized, automatized, and restored with the aid of ANSYS parametric design Language codes. To equip the model with a distinctive ability to encounter various liver models and conditions, pre-operative information that describes the model hyperelasticity and undeformed geometrical parameters will be used to achieve a robust and accurate learning-based force perception model and generalize its applicability for various constitutive models. This study intended to predict the external force through a DNN model. The DNN accepts the invivo deformation point cloud as the input and maps it to the relevant external force. For better performance, dimensionality reduction techniques is deployed to extract the hierarchical features from the input, improving the nonlinear regression accuracy. Deep autoencoder (DAE) has been deployed for downsampling and encoding the input compared to principal component analysis (PCA). Furthermore, this study proposed a novel state vector that includes the geometrical and biomechanical parameters of the liver and the ROI. The DNN hyperparameters were tuned to obtain the highest accuracy. Results highlight a remarkable improvement in the accuracy compared to a model without the state vector. To validate results and their robustness in invivo application, they are compared them with the ground truth obtained from an unseen FE simulation of the liver, knowing that imitating the invivo condition in an experimental setting is a serious barrier. On the other hand, a simplified exvivo evaluation would not assess the proposed model's strength based on the aforementioned reasons.

Chapter 3

Biomechanical Modelling; 3D Reconstruction and Simulation Analysis

3.1 Liver Biomechanics

This study surrogate intra-operative data of liver deformation in minimally invasive surgery with finite element simulation of the organ to enhance our machine with reliable labeled data. Drawbacks in the ex-vivo setting would impact the in-vivo applicability of vision-based models as mentioned in chapter 1. To acquire a comprehensive dataset to imitate the in-vivo characteristics of a hyperelastic tissue, the actual patients' liver organs was designated. The liver is the organ with a significant portion of minimally invasive surgeries operated on [51]. Knowing that the liver is a hyperelastic medium where stress is highly dependent on the strain range, methods that can demonstrate its biomechanical characteristics were assessed to model the tissue accurately and the boundary conditions that impact its dynamic.

3.1.1 Liver Anatomy

The liver is the largest gland (average length of about 28 cm, average height of about 16 cm and average most significant thickness of about 8 cm) in the human body with numerous physiological functions: to filter, metabolize, recycle, detoxify, produce, store, and destroy [52]. It is located in the right hypochondriac and epigastric regions. The liver has a fibrous coat, the so-called Glisson's capsule. Its rheological behavior is quite different from the glandular parenchyma. Five vessel types run through the liver parenchyma: biliary and lymphatic ducts on the one hand and blood vessels (internal portal supply, hepatic arterial tree, and collecting venous network) on the other hand. The portal vein, which conveys blood from the digestive tract to be detoxified and metabolized, is deep into the proper hepatic artery and common bile duct. This region is thus supposed to be wholly stable.

Generally, The right liver extremity is thick and rounded, while the left one is thin and flattened. The anterior border is thin, sharp, and free. The posterior border is connected to the diaphragm by the coronary ligament. The upper surface, covered by the peritoneum, is divided by the suspensory ligament. The lower surface is connected with the gallbladder within the GB fossa, the stomach, the duodenum, the right kidney, and the right part of the transverse colon. These organs are in contact with the liver surface, but they do not interact strongly with the liver; hence they can not be considered a supporting organ. The inferior vena cava travels along the posterior surface, often in a groove. The connection implies another strong fitting condition [53].

3.2 Liver 3D Model Reconstruction

As stated in chapter 1 our 3D model has been constructed based on the computed tomography (CT) DICOM images. The images used in this project came from the 2017 liver tumor segmentation challenge (LiTS)(Christ, 2017), a competition organized by the international conference on medical image computing and computer-assisted intervention (MICCAI 2017) in conjunction with the IEEE international symposium on biomedical imaging (ISBI 2017) whose objectives were the automatic liver segmentation and tumor segmentation, and tumor load estimation. This dataset is publicly available and consists of 130 scans of abdominal CT images from six different medical centers.

This study deployed 3D Slicer software to reconstruct the organ geometries that are used in FEM to provide a broad range of 3D models. 3D Slicer is a powerful visualization tool that allows the exploration of the imaging datasets in two, three, and four dimensions. Slicer enables the fusion of functional and anatomical data and provides a variety of generic and specialized tools for their

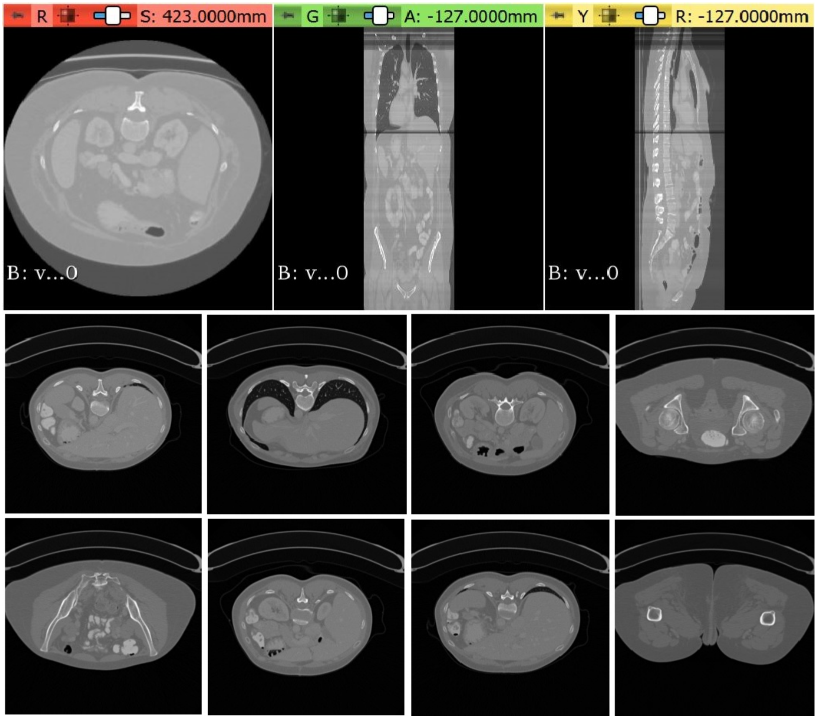


Figure 3.1: CT scan DICOM: liver segmentation extracted for 3D reconstruction

processing, and multi-modal analysis [54]. The generated model in the SRT file format would be prepared (smoothened and warped) afterward, which leads to a finer mesh generation. Twenty-four different geometries have been extracted from patients' DICOM images, and simulation for the various setting has been executed to enhance our training dataset's generalisability (Fig. 3.2).

3.3 Liver Constitutive Model

The robustness of the FE simulations depends on many factors, such as the process of segmentation and the constitutive model that characterizes the mechanical behavior, and last but not least, the boundary conditions. For the modeling of the liver in FE software, first, the liver soft tissue is considered an incompressible, isotropic, hyper-elastic medium, which is characterized by strain energy density function as a part of Helmholtz energy [55]. Neglecting the thermal part, it yields:

$$\sigma^{PK2} = \frac{\partial W}{\partial E} \tag{2}$$

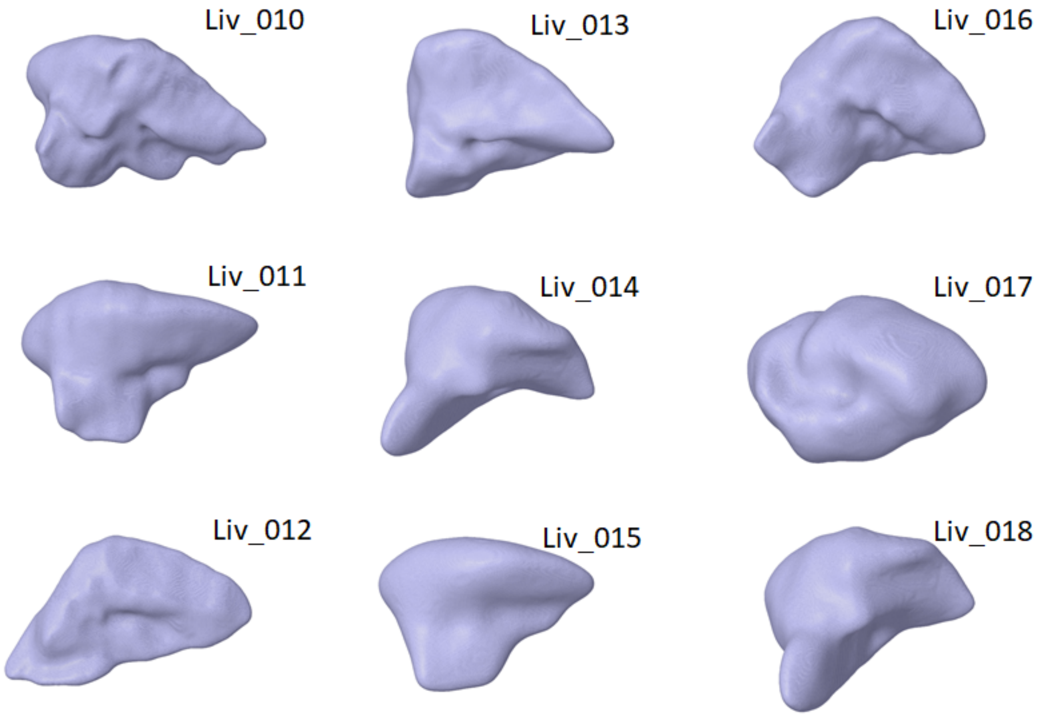


Figure 3.2: Patient specific liver 3d models reconstructed from DICOM images

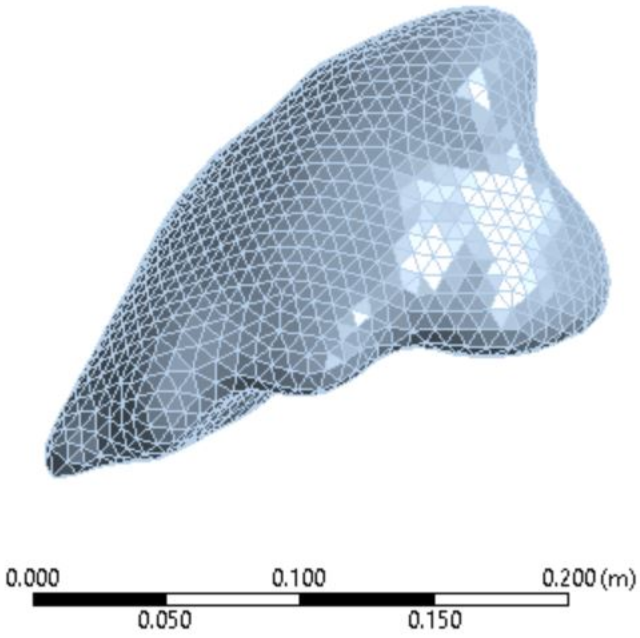


Figure 3.3: Mesh generated for a liver 3D model by ANSYS SpaceClaim

Where W is the strain density function and E is the Green-Lagrangian strain tensor. The stretch ratio, the ratios of initial and final lengths in the principal directions, is defined as:

$$\lambda = 1 + \epsilon_{Eng} \tag{3}$$

Hence, the Green–Lagrange strain tensor is:

$$E_i = (\lambda_i - 1) + \frac{1}{2}(\lambda_i - 1)^2 \tag{4}$$

$$\frac{\partial E_i}{\partial \lambda_i} = \lambda_i \tag{5}$$

Having, Cauchy stress as $\sigma_i = (1 + \epsilon_i)^2 \sigma_i^{PK2}$, we have:

$$\sigma_i = \lambda_i \frac{\partial W}{\partial \lambda_i} \tag{6}$$

The invariant-based continuum mechanics approach defines the strain energy density function for anisotropic, hyperelastic material as a function of invariants. The invariants are computed via the product of the deformation gradient with its transpose.

$$I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \tag{7}$$

$$I_2 = \lambda_1^2\lambda_2^2 + \lambda_2^2\lambda_3^2 + \lambda_3^2\lambda_1^2 \tag{8}$$

$$I_3 = \lambda_1^2\lambda_2^2\lambda_3^2 = J^2 \tag{9}$$

The rate of change of the strain energy density $W(E)$ equals the rate of mechanical work done on the material per unit reference volume.

$$\frac{dW}{dt} = \frac{\partial W}{\partial F_{ij}} \frac{\partial F_{ij}}{\partial t} \tag{10}$$

where:

$$F = I + \nabla \ \times U \tag{11}$$

To obtain a proper biomechanical behavior of tissue, different hyperelasticity models such as Mooney-Rivlin, Ogden, and Yeoh are assigned. We calculated the mean of stress magnitudes relevant to the specified range of strain and applied these models considering the strain energy density function as a function of invariants of the left Cauchy–Green deformation tensor [56]. The strain energy density function for various models are given as follows:

Mooney-Rivlin:

$$W = \sum_i \sum_j C_{ij}(I_1 - 3)^i (I_2 - 3)^j + D(J-1)^2 \tag{12}$$

Ogden:

$$W = \sum_i^N \frac{2\mu_i}{\alpha_i^2}(\lambda_1^{\alpha_i} + \lambda_2^{\alpha_i} + \lambda_3^{\alpha_i} - 3) + \frac{K_1}{2}(J-a)^2 \tag{13}$$

Yeoh:

$$W = \sum_i C_i (I_1 - 3)^i \tag{14}$$

Results as second Piola Kirchhoff stress versus Green Lagrangian strain curve for different ranges of strain rate and the mean values are demonstrated in Fig. 3.4 and Fig. 3.5 [57]. Each models parameter were curvefitted and obtained by ANSYS Structural module and results are shown in Tables 3.1, 3.2 and 3.3 .

A uniaxial tensile test was simulated based on mentioned-above models as shown in Fig. 3.6 and compared the result with that relevant value that is extracted from experiments in [49] studies. Fig. 3.7 shows that the Mooney-Rivlin model results in a more accurate simulation. Hence, five parameter Mooney-Rivlin model was deployed (Eq.15) to simulate the liver's nonlinear hyperelastic behavior. The accuracy metric is NRMSE, and the results are shown in table 3.4

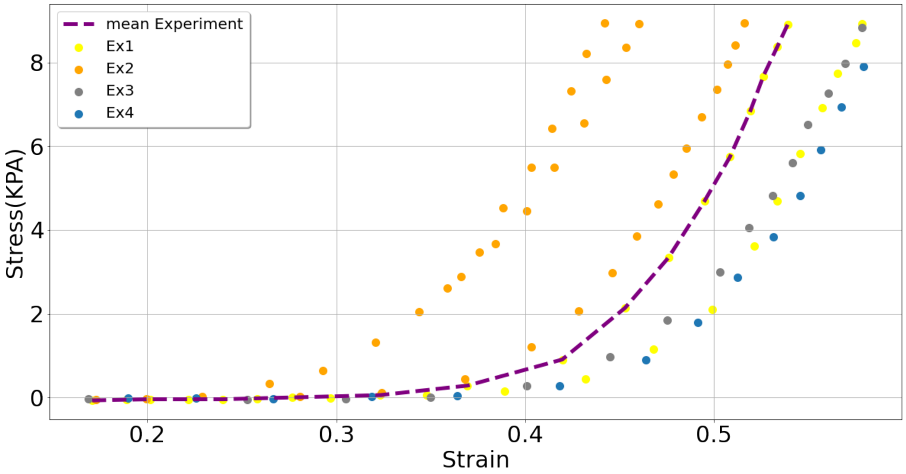


Figure 3.4: Stress-Strain graph of various confined compression experiments

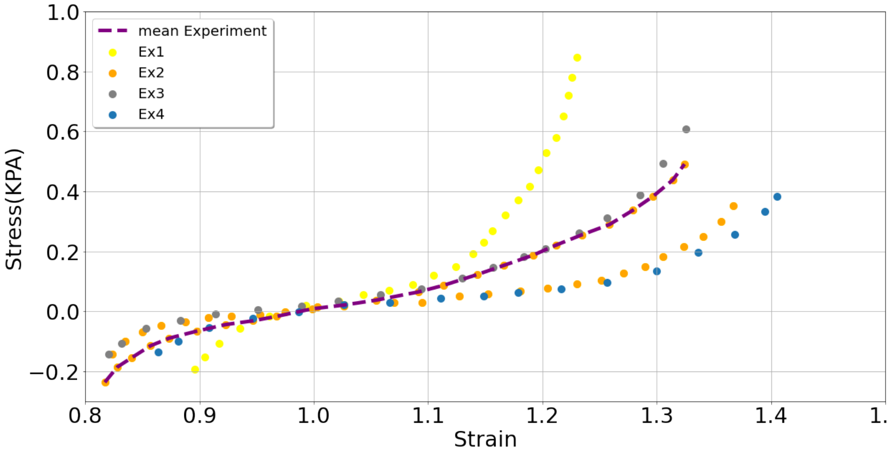


Figure 3.5: Stress-Strain graph of various shear test experiments

Table 3.1: Mooney-Rivlin parameters obtained from from shear and confined compression test

$Components$	$Parameters$	$Values \quad N \times mm^{-4}$
Liver	Density	1.051
	C_{01}	863.5
	C_{10}	-584.5
	C_{11}	3228.7
	C_{20}	-6228.9
	C_{02}	3328.3

Table 3.2: Ogden parameters obtained from shear and confined compression test

$Components$	$Parameters$	$Values \quad N \times mm^{-4}$
Liver	Density	1.051
	μ_1	187.13
	α_1	499.13
	μ_1	10.65
	α_1	15.03

$$\begin{aligned} W = C_{10}\left(\bar{I}_1 - 3\right) + C_{01}\left(\bar{I}_2 - 3\right) + C_{11}\left(\bar{I}_1 - 3)(\bar{I}_1 - 3\right) \\ + C_{20}\left(\bar{I}_1 - 3\right)^2 + C_{20}\left(\bar{I}_1 - 3\right)^2 + \frac{1}{d}\left(\bar{J} - 1\right)^2 \end{aligned} \tag{15}$$

3.4 Boundary Condition

Liver contact with other organs and muscles are considered to imitate the realistic condition of tissue deformation. The stiffness of normal and tangential elastic foundation for coronary and triangular ligament that connects the posterior side of the liver to the diaphragm is considered $56.3 \pm 19.2 N/mm$. The inferior vena cava (IVC) travels along the posterior surface and holds a fixed support condition that constrains the neighborhood nodes. The rib's interaction on the anterior face

Table 3.3: Yeoh parameters obtained from shear and confined compression test

$Components$	$Parameters$	$Values \quad N \times mm^{-4}$
Liver	Density	1.051
	C_{10}	0.00161
	C_{20}	0.0125
	C_{30}	0.00531

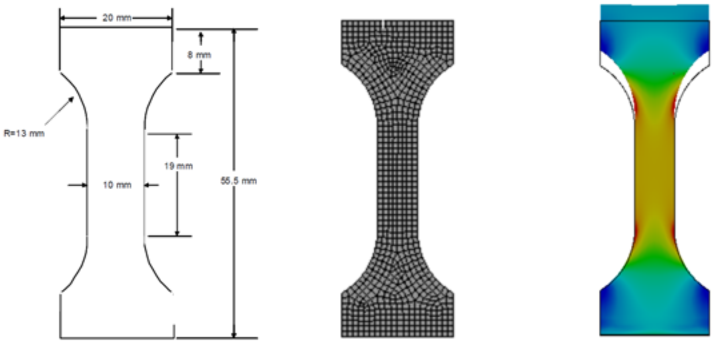


Figure 3.6: Uniaxial test simulation based on different hyper-elastic models

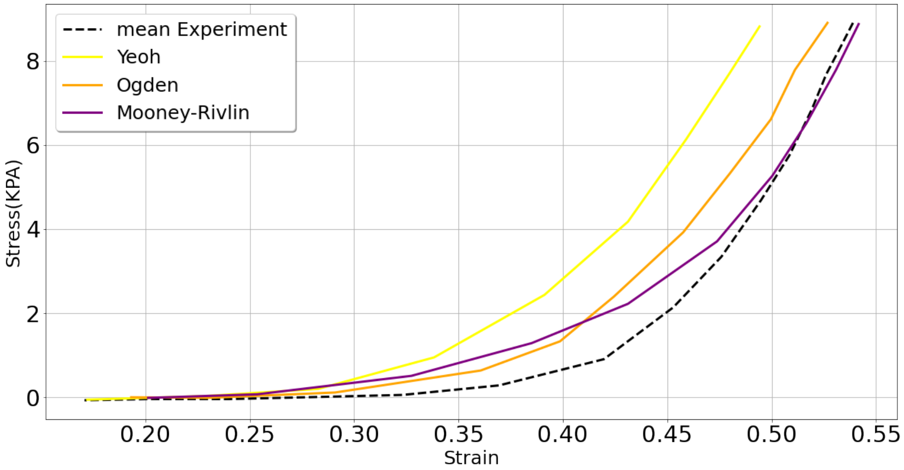


Figure 3.7: Stress-Strain graph of various uniaxial models vs. experimental data

Table 3.4: NRMAE of uniaxial test simulation based off various hyperelastic models

$Mooney-Rivlin$	$Ogden$	$Yeoh$
0.12	0.18	0.21

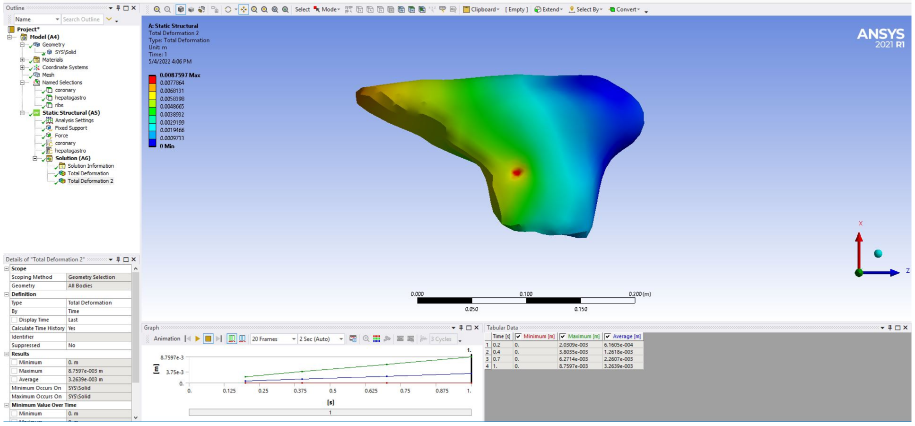
constrains the zone with just tangential motion, so the normal motion is estimated zero [58]. The lower face connection to abdominal organs through hepatoduodenal and hepatogastric ligaments can be estimated as an elastic foundation with a lower stiffness of $42.6 \pm 10.2N/mm$ [59].

3.5 Finite Element Modelling

To consider the impact of the unobserved region in the region of interest study, instead of partial analysis, we assess the entire geometry's biomechanical behavior with the exertion of an external force and the presence of boundary conditions as indicated previously. External surface deformation of ROI can be extracted afterward. ANSYS 2021 R1 software and the static structural module are deployed for this simulation [60]. Seizing and number of meshes vary with regards to the liver geometry. On average, Twenty thousand nodes were generated using tetrahedral mesh. However, the surface nodes were extracted since our observations in minimally invasive surgeries are generally bounded to the surface of ROI.

Twenty-five liver 3D models were studied, and forces in the range of actual MIS were exerted on various positions. In order to realistically analyze the soft tissue, as shown in the previous section Mooney-Rivlin with the lowest normalized root mean squared error has been assigned. It is worth mentioning, the Mooney-Rivlin model does not give any unique insight into material behavior. It is merely curve-fits of various polynomials to test data. The numerical values of coefficients resulting from the curve-fits are entered into FEA programs for use in mechanical analyses. The FEA program knows how stiff the tissue is based on the values of the coefficients. Coefficients were obtained through curve fitting of the nonlinear stress-strain graph of soft tissue experimentation that was grasped from literature.

The Ansys's static structural module is used. Large deflection has been considered with plain strain analysis with auto time stepping and one hundred initial time steps and a maximum of one



Details of "Total Deformation 2"

Scope	
Scoping Method	Geometry Selection
Geometry	All Bodies
Definition	
Type	Total Deformation
By	Time
□ Display Time	Last
Calculate Time History	Yes
Identifier	
Suppressed	No
Results	
□ Minimum	0. m
□ Maximum	8.7597e-003 m
□ Average	3.2639e-003 m
Minimum Occurs On	SYS\Solid
Maximum Occurs On	SYS\Solid
Minimum Value Over Time	
□ Minimum	0. m

	Time [s]	☑ Minimum [m]	☑ Maximum [m]	☑ Average [m]
1	0.2	0.	2.0309e-003	6.1605e-004
2	0.4	0.	3.8035e-003	1.2618e-003
3	0.7	0.	6.2714e-003	2.2607e-003
4	1.	0.	8.7597e-003	3.2639e-003

Figure 3.8: Nodal displacement of liver simulation retrieved on ANSYS 2021R1

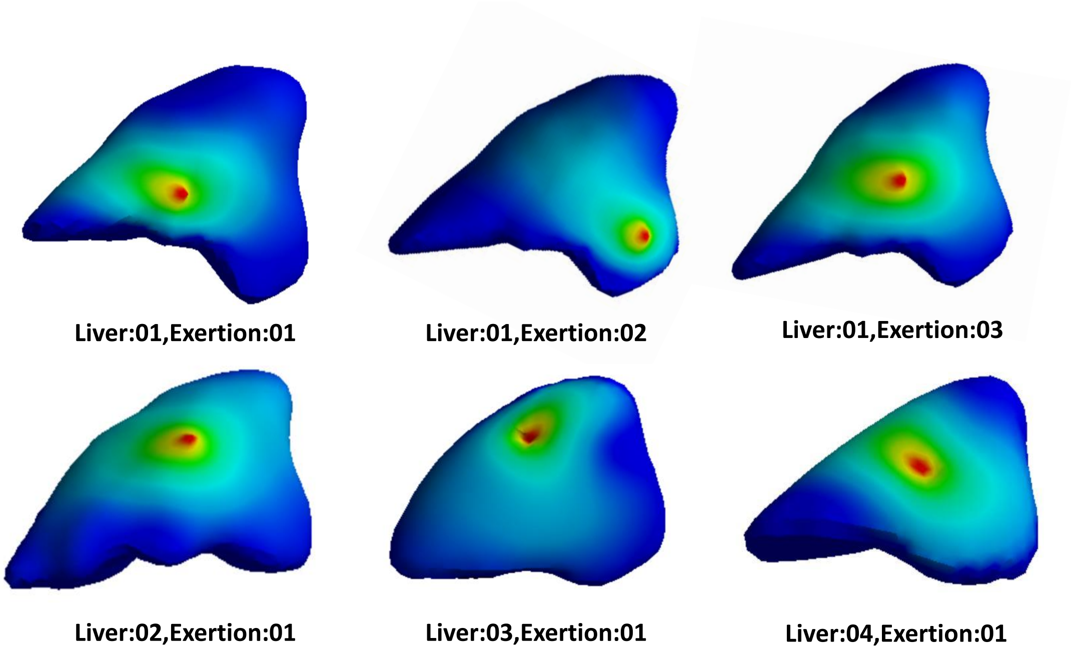
thousand timesteps computed via direct solver. Boundary conditions were imposed on the liver, as illustrated earlier. Since the tangential elastic foundation is not defined based on the boundary's elasticity in the mechanical module, the specified boundary conditions based on the nodal elastic configuration was implemented and defined by the APDL code. APDL is an acronym for ANSYS parametric design language, a powerful scripting language that provides customized automation in preprocessing, solving, and post-processing of ANSYS simulations. APDL is the foundation for features, many of which are not exposed in the workbench mechanical.

The problem is then parameterized, and approximately twenty-four thousand of various sets of force magnitude, exertion point, and liver geometries have been defined. In order to generate sufficient data to be further deployed in deep learning training, each geometrical and elastomechanical simulation setting has been parameterized for thousand sets of various force components. Bellow, the algorithm of the deformation point cloud is shown. Simulations were retrieved from a licensed version of ANSYS 2021 R1 in Concordia University. Fig 3.8 demonstrates the setting of Ansys mechanical for the simulation mentioned above.

APDL commanding is used to automate retrieving simulations and restoring them in a specified directory. This powerful scripting tool enables us to generate as much data as required to implement deep learning accurately and accordingly. Results have been retrieved and exported as samples are shown in Fig. 3.9. For each geometry and selected exertion coordinate, Force components are defined randomly relevant to the range of forces in actual surgeries. For each setting, thousands of

Algorithm 1 ROI deformation extraction Algorithm

```
  Initialize $g_i^L, p_i^e, E(s)$
  ${C^L}_i \leftarrow E(s)$
Require: $D(x, y, z)$, Design point comply the hyper-elasticity criterion
  for $I = 1 : N$ do
      while $p \geq 1e - 7$ do
          Compute stiffness Matrix K
          $-p. = \Delta KU$
          $u \leftarrow u + \Delta u$
          Compute residual p
      end while
      for $I = 1 : 3$ do
          Set, displacement, U, $X_i$
          Get, numnode, node, count
          Get, nd_sig(L), node, U, $X_i$
          Open, 'Dis_i' , CSV
          Write, nd_sig(L), (E13.6)
      end for
      for $i = 1, S$ do
          if $d(Dis_i, p^e) \leq Threshold$ then
              $U \leftarrow Roi\_i, X_i$
          else Pass
          end if
      end for
  end for
```



Liver:01,Exertion:01

Liver:01,Exertion:02

Liver:01,Exertion:03

Liver:02,Exertion:01

Liver:03,Exertion:01

Liver:04,Exertion:01

Figure 3.9: FEA simulation for various liver model, various hyperelasticity and exertion position

simulations were executed, parametrized, and stored automatically. It is worth mentioning that the range of force components is chosen to comply with the continuum hyperelastic behavior of the medium [61].

Chapter 4

Data Processing and Deep Neural Network

4.1 Deep Learning

Machine Learning systems have been proven to be a robust tool for numerous applications, from identifying objects in images to speech-to-text transcription, matching news, items, posts, or products relevant to user's preference, and selecting desired search results. Increasingly, these applications use a class of techniques called deep learning. Deep learning provides computational models composed of multiple processing layers that, as Lecun stated, learn a representation of data with multiple abstraction levels that are effectively designated for speech recognition, vision-based recognition, object detection and segmentation, drug discovery, and genomics. Deep learning discovers intricate structures in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer.

Multilayer perceptron (MLP) consists of input, hidden, and output layers. Each layer comprises neurons that acquire an activation function. An activation function is a function in an artificial neuron that delivers an output based on inputs. Activation functions in artificial neurons are an essential part of the role of artificial neurons in modern artificial neural networks that resemble a biological neuron's function. A deep-learning architecture is a multilayer stack of simple modules, all (or most) of which are subject to learning, and many compute nonlinear input-output mappings. Each module in the stack transforms its input to increase both the selectivity and the invariance of the representation. With multiple nonlinear layers, say a depth of 2 to 20, a system can implement incredibly intricate functions of its inputs that are simultaneously sensitive to minute details, as shown in Fig. 4.1.

Backpropagation computes the gradient of the loss function concerning the network's weights for a single input-output example. It does so efficiently, unlike a naive direct computation of the gradient for each weight individually. This efficiency makes it feasible to use gradient methods for training multilayer networks, updating weights to minimize loss; gradient descent, or variants such as stochastic gradient descent, are commonly used. The backpropagation algorithm works by computing the gradient of the loss function for each weight by the chain rule, computing the gradient one layer at a time, iterating backward from the last layer to avoid redundant calculations of the intermediate terms chain rule. Other intermediate quantities are used in the derivation of backpropagation; they are introduced as needed below. Bias terms are not treated specially, as they correspond to a weight with a fixed input of 1. For the purpose of backpropagation, the specific loss function and activation functions do not matter as long as they and their derivatives can be evaluated efficiently.

$$g(x) = f^L(W^L F^{L-1}(W^{L-1}...f^1(W^1x)...) \tag{16}$$

4.2 Proposed Method

Our method is a learning-based force estimation implemented by a deep neural network. In order to diminish the noise and perturbation that occurs in the image-based CNN, a deformation point cloud was used, which in practice can be acquired from stereoscopic images of surgery through existing 3D reconstruction techniques and depth imaging [62]. Fig. 4.2 demonstrate the procedure schematically.

Our project concentrates on the model that takes surface 3D point cloud as an input with pre-operative information of the patient organ and predicts the magnitudes and directions of the forces.

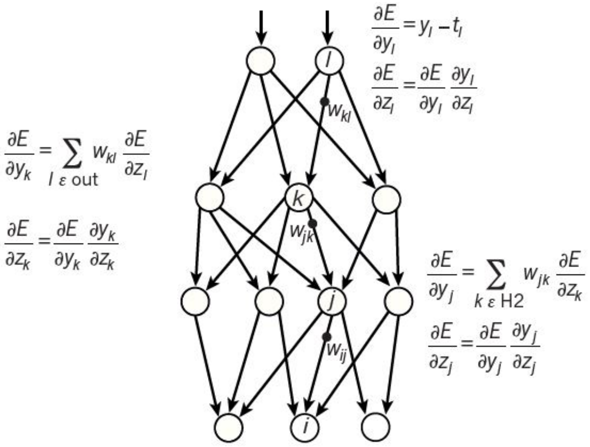

$\frac{\partial E}{\partial y_l} = y_l - t_l$
$\frac{\partial E}{\partial z_l} = \frac{\partial E}{\partial y_l} \frac{\partial y_l}{\partial z_l}$
l
w_{kl}
$\frac{\partial E}{\partial y_k} = \sum_{l \, \varepsilon \, \text{out}} w_{kl} \frac{\partial E}{\partial z_l}$
$\frac{\partial E}{\partial z_k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial z_k}$
k
w_{jk}
$\frac{\partial E}{\partial y_j} = \sum_{k \, \varepsilon \, \text{H2}} w_{jk} \frac{\partial E}{\partial z_k}$
$\frac{\partial E}{\partial z_j} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial z_j}$
j
w_{ij}
i

Figure 4.1: Back propagation optimization in Neural networks

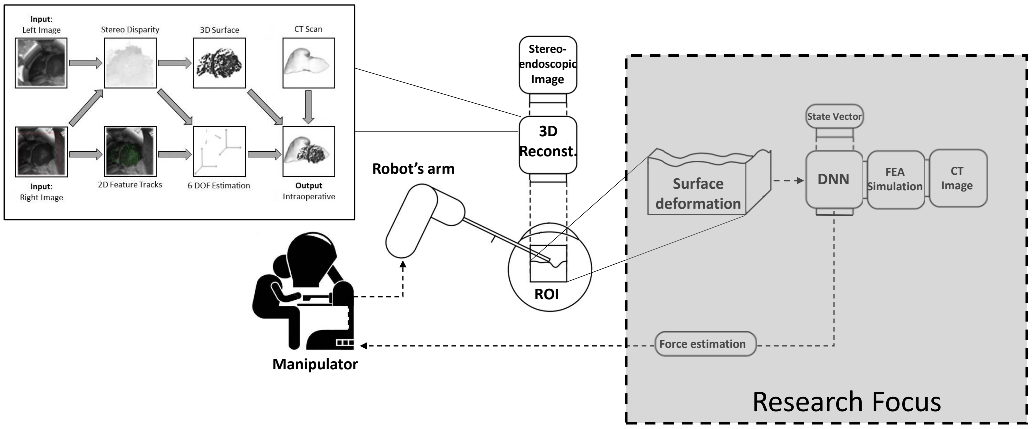


Figure 4.2: Proposed model schematic, learning-based DNN with pre-operative state vector

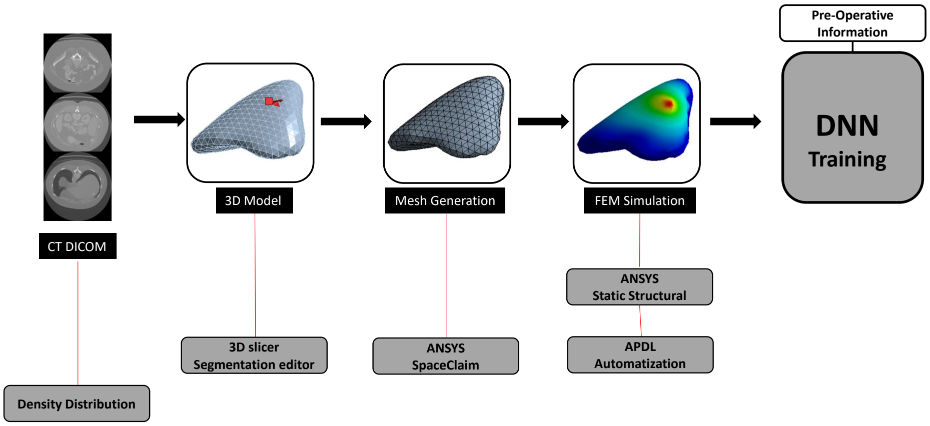


Figure 4.3: Synthetic dataset generated from digital twin of patient specific liver 3D model

This model has been trained on the surface point cloud obtained from FEA simulation as a digital twin of actual in-vivo hyperelastic deformation of live soft tissue. In practice, this point cloud can be extracted from MIS stereoscopic images. As shown in Fig. 4.3 the synthetic dataset has been generated via FEA simulation of a patient-specific liver 3D model that would be extracted and reconstructed from CT scan images. The reconstruction procedure is out of the scope of our research and has been addressed and studied remarkably in the literature.

It is also worth mentioning that technologies such as time of flight can be designated to acquire a 3D depth mapping of MIS surroundings, and recently researchers have proposed its application in surgical sites [63]. Therefore the 3D point cloud obtained from our FEA simulation on various patient's actual liver geometries would surrogate the reconstructed 3D point cloud of the actual liver in MIS.

The model is trained based on a 3D FEA simulation of the patient-specific liver model. The surface deformation point cloud has been used as the input of our DNN. A portion of the surface that is typically observable in MIS images, AKA, region of interest, is used in our method, which complies with the actual limitation of surgeries. Fig. 4.4 pictures the procedure step by step.

This study's models are trained based on four different scenarios to assess the DNN models' generalisability and investigate various features' impact on prediction accuracy. The first scenario's

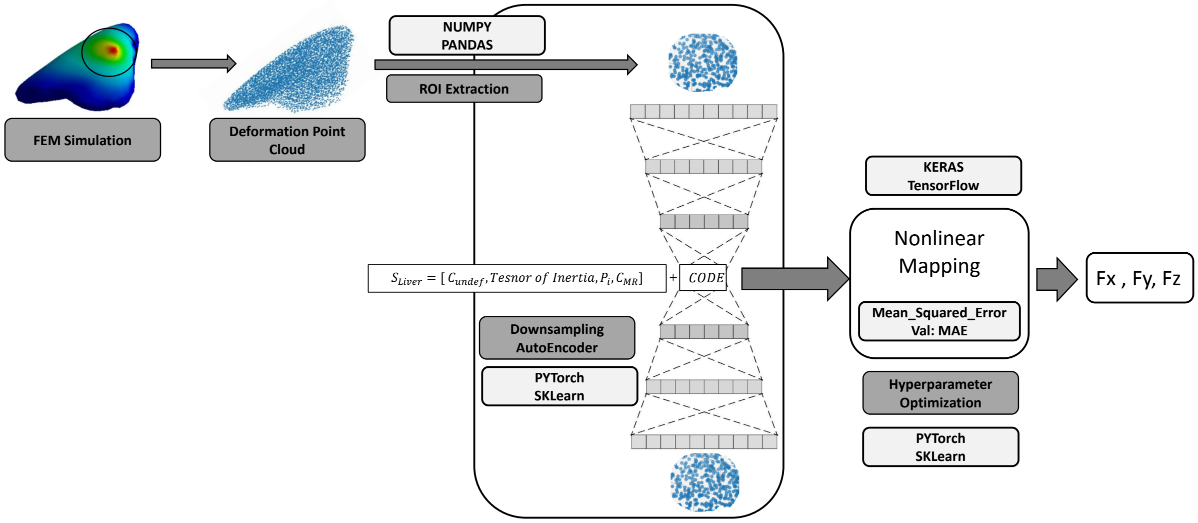


Figure 4.4: Proposed model; simulation pointcloud input to the deep neural network

dataset comprises one liver geometry with a fixed exertion point and identical hyperelasticity. In the second scenario, geometry and hyperelasticity are identical, but the exertion position and the force components are variable. The third scenario includes a variable geometry, unlike the previous one. Hyperelasticity is the only constant term in this dataset and the fourth scenario in which all mentioned parameters vary. We have included pre-operative information in our model to consider these characteristics. A state vector comprises liver geometry undeformed state, exertion, centroid position, and hyperelasticity.

$$S = S(C^{undef}, C_{ROI}^{undef}, C_{ROI}^{def}, I_{x,y,z}, P, C_i) \tag{17}$$

Where C^{undef} is the undeformed liver model centroid. C_{ROI}^{undef} and C_{ROI}^{def} are undeformed and deformed ROI centroids, respectively. $I_{x,y,z}$ is the second moment of inertia for a global axis and can be written as :

-I$_{yx}I_{yy} - I_{yz}$

-I$_{zx} - I_{zy}I_{zz}$

In the proposed state vector, P is the force exertion position, and C_is are Mooney Rivlin's constants of hyperelasticity that vary in different tissues and geometries.

$$I_{xx} = \int_m (y^2 + z^2) dm I_{xy} = \int_m (xy) dm I_{yy} = \int_m (x^2 + z^2) dm I_{xz} = \int_m (xz) dm I_{zz} = \int_m (x^2 + y^2) dm I_{yz} = \int_m \tag{18}$$

P is the force exertion position that can be the mean value of all nodal coordinates, and finally, C_i that are Mooney Rivlin constants of hyperelasticity that vary in different tissues and geometries. The state vector is then concatenated to the latent representation of deformed geometry on the surface of the region of interest. Various surface deformation portions have been studied to culminate in the highest accuracy.

The downsampled code was added to the state vector and was nonlinearly mapped to the force components, F_x, F_y, F_z. A multilayer fully connected Perceptron has been used to execute the nonlinear regression. For each scenario based on the dataset, various hyperparameters have been used. These parameters are optimized based on a hyperparameter optimization algorithm executed by the AX API package in Python. We run a total of 200 trials, and each trial evaluates the possible combinations of hyperparameter values and spits out the scores output. Results will be discussed in the chapter 5.

This process keeps track of the history of parameters and scores and makes an intelligent guess of the following better set of parameters. The rectifier activation function (Relu) is used, and the Adam optimizer and RMSPROP, the most popular optimizer based on the stochastic gradient descent approach, are deployed. Adam can be looked at as a combination of RMSprop and stochastic gradient descent (SGD) with momentum. It uses estimations of the first and second moments of the gradient to adapt the learning rate for each neural network weight. Two performance metrics were used to train and evaluate the accuracy of the DL model, including mean squared error (MSE) and mean absolute error (MAE). For each deformation caused by an external force, the MSE is defined by Eq. (19).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(F_i - \hat{F}_i)^2 \tag{19}$$

Where F_i is the external force component, MAE is defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(F_i - \hat{F}_i) \tag{20}$$

4.3 Data Preprocessing

Our model's input consists of spatial coordinates of liver surface deformed state with respect to a global coordinate system. Each node's displacement was retrieved from the ANSYS simulation and added to its undeformed components for the same coordinate system.

$$X_{def} = X_{undef} + \Delta X \tag{21}$$

This input will then be standardized and normalized. Standardization of datasets is a common requirement for many machine learning estimators. Due to our input order of magnitudes, this step is irreplaceable since some features vary in orders of magnitude and are larger than others. It might dominate the objective function and make the estimator unable to learn from other features correctly as expected. Furthermore, to provide an input independent of the coordinate origin, this step is essential.

StandardScaler method from the Scikit-learn library pre-processing class in Python was used to standardizes features by removing the mean and scaling to unit variance. The standard score of sample X is calculated as:

$$Z = (X - U)/S \tag{22}$$

where U is the mean of the training samples or zero if with_mean=False and S is the standard deviation of the training samples or one if with_std=False.

Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set. Mean and standard deviation are then stored to be used on later data using the transform function.

4.4 Downsampling and Dimensionality Reduction

A legitimate, adequate data set is indeed required to implement feed-forward deep learning properly. As indicated in chapter 1, our proposed method comprises of dimensionality reduction part to reduce the relatively extensive size input of the deformation point cloud. This study investigated different powerful techniques such as principal component analysis (PCA) and auto-encoder to execute accurate and effective downsampling that retains the most important features of the input vector for a precise force estimation. Practically speaking, our model uses partial surface nodal coordinates of deformed geometry that is acquired from the FEA simulation. The input is downsampled via mentioned techniques, and the outcome is compared in chapter 5. In the following, the implementations is discussed in detail.

4.4.1 Principal Component Analysis

Principal component analysis (PCA) is a hierarchical coordinate system based on singular value decomposition [64]. The PCA is widely used in statistical shape analysis and is chosen as this study's surface deformation encoding method. Principal components analysis (PCA) is one of the central uses of the SVD, providing a data-driven, hierarchical coordinate system to represent high-dimensional correlated data. This coordinate system involves the correlation matrices. Importantly, PCA pre-processes the data by mean subtraction and sets the variance to unity before performing the SVD. The geometry of the resulting coordinate system is determined by principal components (PCs) that are uncorrelated (orthogonal) to each other but have a maximal correlation with the measurements. Matrix of X can be decomposed as:

$$X = U\Sigma V^T \tag{23}$$

$$X = \begin{bmatrix} \cdots & x_1 & \cdots \\ \vdots & \ddots & \vdots \\ \cdots & x_n & \cdots \end{bmatrix}$$

$$\bar{x} = \frac{1}{n}\sum_{j=1}^{n} x_j$$

Figure 4.5: will change to equation later

Where U is left singular vector $\epsilon C^{(nn)}$, V is right singular vectors $\epsilon C^{(mm)}$ that are orthogonal and is a diagonal matrix $\epsilon C^{(mm)}$. For Matrix X with rather massive amount of data, X , the mean matrix would be defined as:

The covariance matrix can be defined as a product of the mean subtract matrix and its transpose:

$$C = B^T.B \tag{24}$$

Where:

$$B = X - \tilde{X} \tag{25}$$

Principle component matrix T will be defined as:

$$T = B.W \tag{26}$$

Where W is the eigenvector of the covariance matrix. Knowing that eigenvalue of covariance matrix equal singular value square and W is orthogonal, it yields;

$$X - \bar{X} = \sum_{m=1}^{M} \alpha_m.\sqrt{\gamma_m}.{W_m}^T \tag{27}$$

M can be chosen from one to any principal component direction that suffices approximation accuracy. The feature scaling and normalization of data and implementation of PCA are computed through Scikit-learn Library in Python, which uses the LAPACK implementation of the full SVD to provide routines for solving systems of singular value problems. The hierarchical coordinates

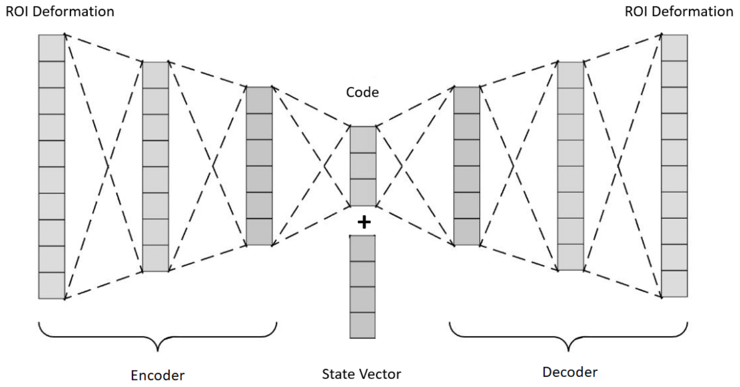


Figure 4.6: Autoencoder neural network architecture

demonstrate data distribution for each direction. Surface deformation encoding can be depicted as a neural network with linear activation, and each link's weight is equal to the relevant components of eigenvectors, scaled by relative root square eigenvalue, $w_m/\sqrt{\lambda_m}$.

4.4.2 AutoEncoder

Autoencoder is one type of feedforward neural networks that comprises of encoding and decoding sections. The code or downsampled section as shown in fig 4.6 is a representation of important features that describe the input and output characteristics.

Input is encoded through layers and decoded to the same output afterward. The code is a compact ”summary” or ”compression” of the input, also called the latent-space representation.

Autoencoders are data-specific, meaning they will only be able to compress data similar to what they have been trained on. It is lossy, which means that the decompressed outputs will be degraded compared to the original inputs and they are learned automatically from data examples, which is a valuable property: it means that it is easy to train specialized instances of the algorithm that will perform well on a specific type of input.

As illustrated above, The input, which in our case is simulated deformation of the liver surface point cloud, passes through the encoder, which is a fully-connected ANN, to produce the code. The decoder, which has a similar ANN structure, produces the output only using the code. The goal is to get an identical output similar to the input. Note that the decoder architecture is the mirror image of the encoder. This is not a requirement, but it is typically the case. The only requirement is that the dimensionality of the input and output needs to be the same. Anything in the middle can be played with. In AutoEncoder implementation, the Sigmoid activation function is traditionally used in most cases. In the sigmoid activation function, also called the logistic function, the input is transformed into a value between 0.0 and 1.0. Inputs larger than 1.0 are transformed to the value 1.0. Similarly, values much smaller than 0.0 are snapped to 0.0. The shape of the function for all possible inputs is an S-shape from zero up through 0.5 to 1.0. A general drawback is that they saturate. This means large values snap to 1.0 and small values snap to zero. Hence it is most sensitive to changes around the mid-point of their input, such as 0.5 for sigmoid. The limited sensitivity and saturation of the function happen regardless of whether the summed activation from the node provided as input contains valuable information or not.

Rectified linear unit (ReLU) and its extension Leaky ReLU were tested. Many advantages have been detected by using leaky ReLU, such as computational simplicity and representational sparsity, which means it can output a genuine zero value. The leaky rectifier allows for a slight, non-zero gradient when the unit is saturated and not active. This extension aims to fix the ”dying ReLU” problem. Instead of the function being zero when x < 0, a leaky ReLU will have a slight positive slope. Hence the function computes

$$f(x) = (x < 0)(\alpha x) + 1(x \geq 0)(x)f(x) = (x < 0)(\alpha x) + 1(x \geq 0)(x) \tag{28}$$

Denoising AutoEncoder

In order to consider uncertainty due to the surgical tool occlusion that impacts the accuracy of the 3D reconstruction procedure, an intentional noise was imposed on input, surrounding the exertion point to imitate the presence of the surgical tool and denoised the input via a denoising autoencoder model as illustrated in Fig. 4.7 Results concerning the presence of tool occlusion were discussed in the following chapter.

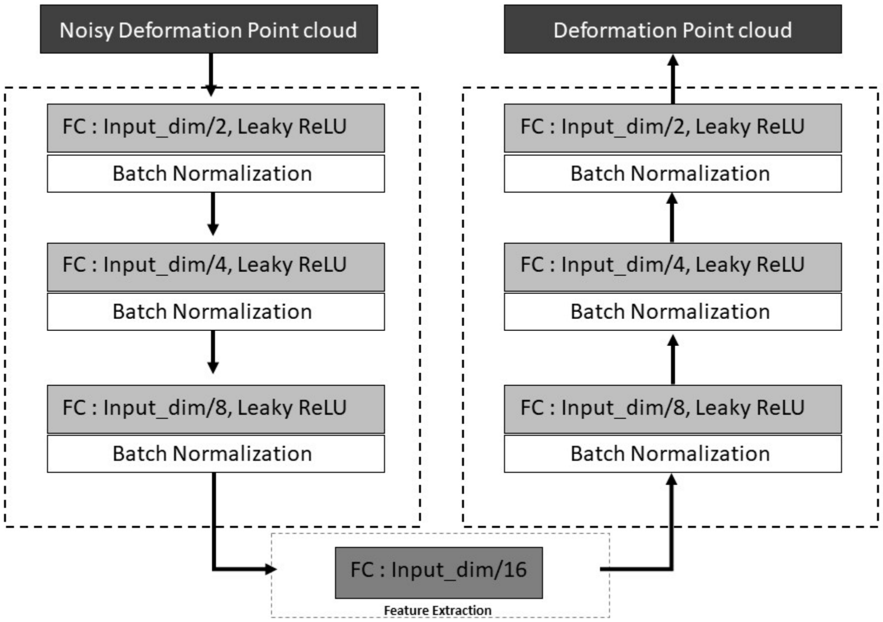

Noisy Deformation Point cloud
FC : Input_dim/2, Leaky ReLU
Batch Normalization
FC : Input_dim/4, Leaky ReLU
Batch Normalization
FC : Input_dim/8, Leaky ReLU
Batch Normalization
FC : Input_dim/16
Feature Extraction
FC : Input_dim/8, Leaky ReLU
Batch Normalization
FC : Input_dim/4, Leaky ReLU
Batch Normalization
FC : Input_dim/2, Leaky ReLU
Batch Normalization
Deformation Point cloud

Figure 4.7: Denoising autoencoder neural network architecture used in the proposed model

4.5 Nonlinear Regression

The code extracted from DAE is then added to the state vector. Through a fully connected multilayer perceptron, this flattened vector has been mapped to the nodal force magnitudes. DL is an optimization problem, and a set of hyperparameters needs to be determined for higher accuracy, i.e., lower error. For this purpose, many techniques are introduced, such as grid search and random search to optimize the ANN parameters such as the number of hidden neurons and layers, learning rate, dropout, etc. Adam optimizer was used to train the model. PyTorch API AX package is used for tunning hyperparameters [65]. For any set of given hyperparameter values, this function returns the mean and standard deviation of the error based on the ten-fold cross-validation evaluation.

The implementation algorithm based on the proposed model is shown in Algorithm 2. The ROI deformation point cloud is fed to the DNN model. Features are extracted via autoencoder with leaky relu activation and batch normalization at each layer, then concatenated by the state vector. It is mapped to the external force components. Optimizer is implemented Via Keras Library, and arguments are defined as below:

- learning_rate: A Tensor, floating point value, or a schedule that is a tf.Keras.optimizers.schedules. LearningRateSchedule, or a callable that takes no arguments and returns the actual value to use The learning rate. Defaults to 0.001.
- beta_1: A float value or a constant float tensor, or a callable that takes no arguments and returns the actual value. The exponential decay rate for the first moment is estimated default to 0.9.
- beta_2: A float value or a constant float tensor, or a callable that takes no arguments and returns the actual value, The exponential decay rate for the second moment estimates. Defaults to 0.999.
- epsilon: A small constant for numerical stability. This epsilon is ”epsilon hat” in the Kingma and Ba paper (in the formula just before Section 2.1). Defaults to $1e^{-7}$.
- amsgrad: Boolean. Whether to apply AMSGrad variant of this algorithm from the paper ”On the Convergence of Adam and beyond”. Defaults to False.
- name: Optional name for the operations created when applying gradients. Defaults to ”Adam”.
- **kwargs: Keyword arguments. Allowed to be one of ”clipnorm” or ”clipvalue”. ”clipnorm” (float) clips gradients by norm; ”clipvalue” (float) clips gradients by value.

4.6 Hyperparameter Optimization and Cross Validation

The solution hypothesized by deep learning allows computers to learn from experience and understand the phenomena in terms of a hierarchy of concepts [66], with each concept defined in terms of its relation to more straightforward concepts. Hyperparameters are the main parameters that must be defined before the training step, while model parameters will be learned through training procedures. Deep learning is an optimization problem, and a set of hyperparameters needs to be determined to result in higher accuracy, i.e., lower loss. For this purpose, many algorithms can be

Algorithm 2 Force nonlinear mapping from downsampled point cloud deformation concatenated by state vector.

```
Require: Roi_D $\epsilon R^3$, Deformation point cloud for region of interest.
  D = number of simulations
  DL $\epsilon[AE, MLP]$, Where AE: AutoEncoder, MLP: Multi-Layer Perceptron
  $e_{DL}$ = Number of epochs
  $b_{DL}$ = Batch size
  $L_{DL}$ = learning rate
  $\theta_{DL} = [w, b_i, h_i]$
  $O_{DL} = [O_1, ..., O_l]$   where $O_{DL,i} = [O_{i,1}, ..., O_{i,n}]$
  Code = AE Training $(e, b, X, c, l, \theta)$
  for $i = 1 : l$, Where l: Number of autoencoder hidden layers do
    for $j = 1 : len(O_l)$, Where $len(O_l)$: Number of autoencoder hidden neurons in Layer l do
      $O_{1,j}$= Leaky_ RELU $(X * W + b_j)$
      $O_{i,j}$= Leaky_ RELU $(O_{i-1} * W + b_j)$
      Code = AE Training (e,b,Roi,l,Optimizer='ADAM')
    end for
  end for
  S=[$C_{undef}, ROI_{undef}, ROI_{def}, I_{xyz}, p_e, C_{MR}$]
  Code $\leftarrow$ Code + S
  for $i = 1 : l$, Where l: Number of MLP hidden layers do
    for $j = 1 : len(O_l)$, Where $len(O_l)$: Number of MLP hidden neurons in Layer l do
      $O_{1,j}$= RELU $(X * W + b_j)$
      $O_{i,j}$= RELU $(O_{i-1} * W + b_j)$
      Force = MLP Training ($e_{MLP}, b_{MLP}$,Code,l,Optimizer='ADAM')
    end for
  end for
```

used, such as Grid Search and Random Search, to optimize the ANN parameters such as the number of hidden neurons and layers, learning rate, dropout, etc. PyTorch API AX package is utilized. AX (Adaptive Experience) is an open-source package from PyTorch that helps find minima for any function over the range of defined parameters. It is commonly used in ML and DL to optimize the hyperparameters that result in a minimum loss. The optimization is applied by running multiple runs of training, each with a different set of parameters and returning ones with the lowest loss. Ax finds minimas for both continuous parameters such as learning rate and discrete parameters such as the size of a hidden layer. For any set of given hyperparameter values, this function returns the mean and standard deviation of the score (MSE) based on cross-validation or CV score. It uses Bayesian optimization for the former and bandit optimization for the latter. Bayesian optimization provides a principled technique based on the Bayes Theorem to direct a search for a global optimization problem that is efficient and effective. It works by building a probabilistic model of the objective function, called the surrogate function, that is then searched efficiently with an acquisition function before candidate samples are chosen to evaluate the actual objective function. Even though the package is from Pytorch, it will work for any function as long as it returns a single value that needs to be minimized. these parameters are:

- Number of hidden Layers
- Number of neurons per layer
- Learning rate
- Activation function
- Optimizer settings

Chapter 5

Results and Evaluation

5.1 Deformation of Soft tissue

Our model utilizes the 3D depth map, and the deformation is mapped to force magnitudes considering preoperative state parameters as explained thoroughly. To compare the accuracy of devised model, knowing the fact that almost all of the recently proposed models in the literature are trained based on labeled data that are obtained experimentally (ex-vivo setting), this model is trained based on a simulation that imitates the experimental setting of the study in [45].

5.1.1 Surgical tools' occlusion

Apart from the deviation that may occur due to changes in camera and lighting setting and their impact on the stereoscopic image and the prediction, respectively, one significant noise can be the presence of the tool itself, which in our model will cause a perturbation in 3D reconstruction. Hence as discussed previously, the closest nodes to the exertion position were eliminated from our machine input, and results have been observed and compared with the ground truth. Fig. 5.1 demonstrates the predicted force applied on an artificial cubic phantom with and without the presence of tool noise.

5.1.2 Region of Interest Impact

To extend the invivo compatibility, our model utilizes the surface point cloud of the patient-specific liver model as the machine input, which eventually will be mapped to force components

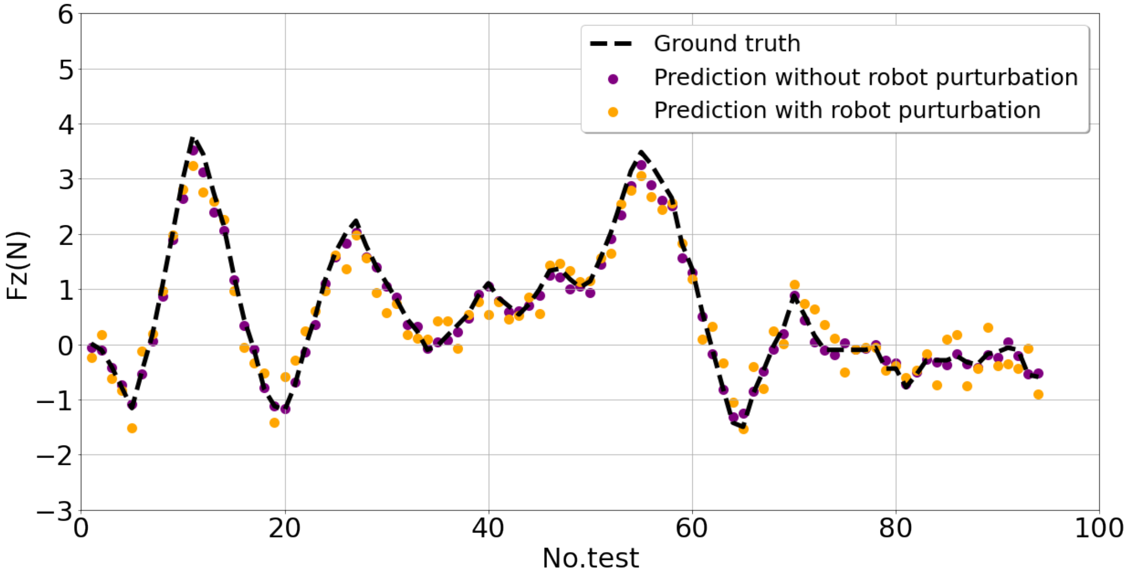


Figure 5.1: Force prediction for the experimental setting obtained from [67] with surgical tool perturbation

relevant to the deformation. It is worth mentioning that not the whole surface of the organ would be assessed, and it can be shown that it is indeed unnecessary to deploy the whole surface. A portion of the exertion position's surface would suffice a reasonably accurate prediction. Fig. 5.2 shows the model's accuracy for various portions of the region of interest (ROI). It is discernible that eighteen percent of surface area in the exertion position neighborhood would bring sufficient accuracy, and adding more surface will not increase the accuracy remarkably.

To assess our proposed model's robustness, accuracy, and generalisability, four scenarios based on synthetic data were devised, where our model is trained, considering the variability of exertion point, hyperelastic properties, and liver geometries.

5.2 Scenario I : Fixed Exertion Point, Identical Geometry, Identical Hyperelasticity

A liver 3D model has been reconstructed from CT images and studied with a single exertion point and variable force magnitudes and direction. Based on Bayesian optimization, hyperparameters are tuned using PyTorch's AX package, and a total number of 200 trials has been executed as

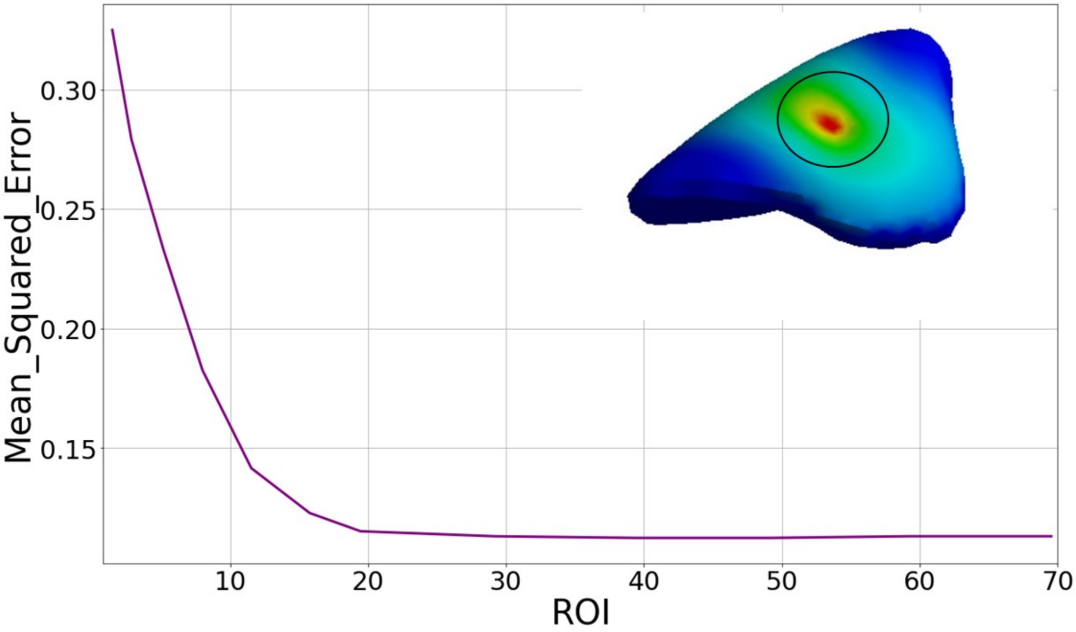


Figure 5.2: Model accuracy as a function of ROI percentage

shown in table 5.1.

Ten fold cross validation score has been monitored for each trial as shown in Fig. 5.3. Table 5.2 specify the results with least CV score, i.e. highest accuracy.

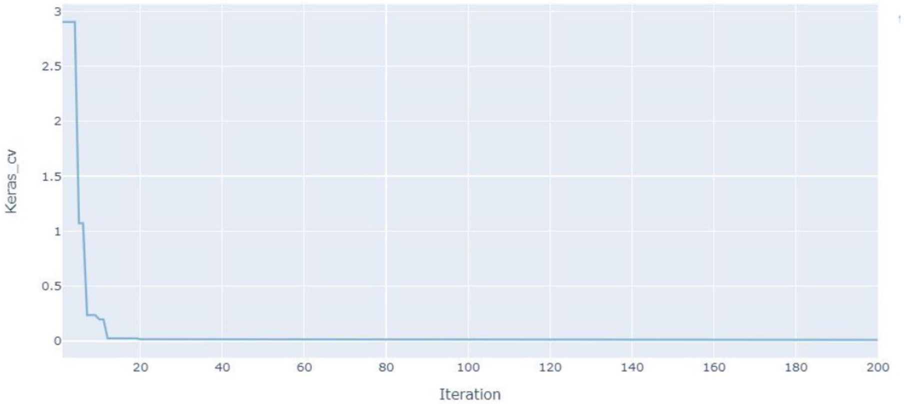


Figure 5.3: Hyperparameter tunning with AX package powered by PYTORCH for scenario I

Fig. 5.4 shows the evaluation and training error through training epochs, and it confirms the

Table 5.1: Hyperparameters optimization results with highest CV score

$CV score$	LR	DO	HN	HL	BS
0.0153	0.0013	0.0133	2	128	32
0.0371	0.0002	0.0200	2	76	16
0.0416	0.0135	0.0247	3	291	64
0.0379	0.0003	0.0105	3	283	256
0.0270	0.0013	0.0133	2	271	64
0.0346	0.0132	0.0207	2	64	64
0.0335	0.0001	0.0187	2	290	16
0.0198	0.0027	0.0032	2	70	16
0.0314	0.0158	0.0140	3	20	64
0.0228	0.0187	0.0142	3	152	32

LR = Learning rate
do = drop out
HL = Hidden Layer
HN = Hidden Neurons
BS= Batch size

Table 5.2: Hyperparameter with lowest CV score

Learning rate	0.0014602
Dropout rate	0.0152328
Hidden Layers	4
Neurons per layer	175
Batch size	64
Keras CV	0.067

convergence of training, which is not over or under fitted. Fig. 5.5 shows the predicted force components compared to the ground truth for scenario I and 5.6 shows the deviation of force prediction from target value.

5.3 Scenario II : Variable Exertion Point, Identical Geometry, Identical Hyperelasticity

Variable force exertion point on a liver 3D model has been studied, and numerous FEA simulations have been retrieved for the model's training. Table 5.3 shows the hyperparameter tuning and Fig. 5.7 pictures the CV score for 200 trial. Table 5.4 shows the optimal combination. Fig. 5.8 shows the training SGD optimization over the number of epochs, i.e., training history.

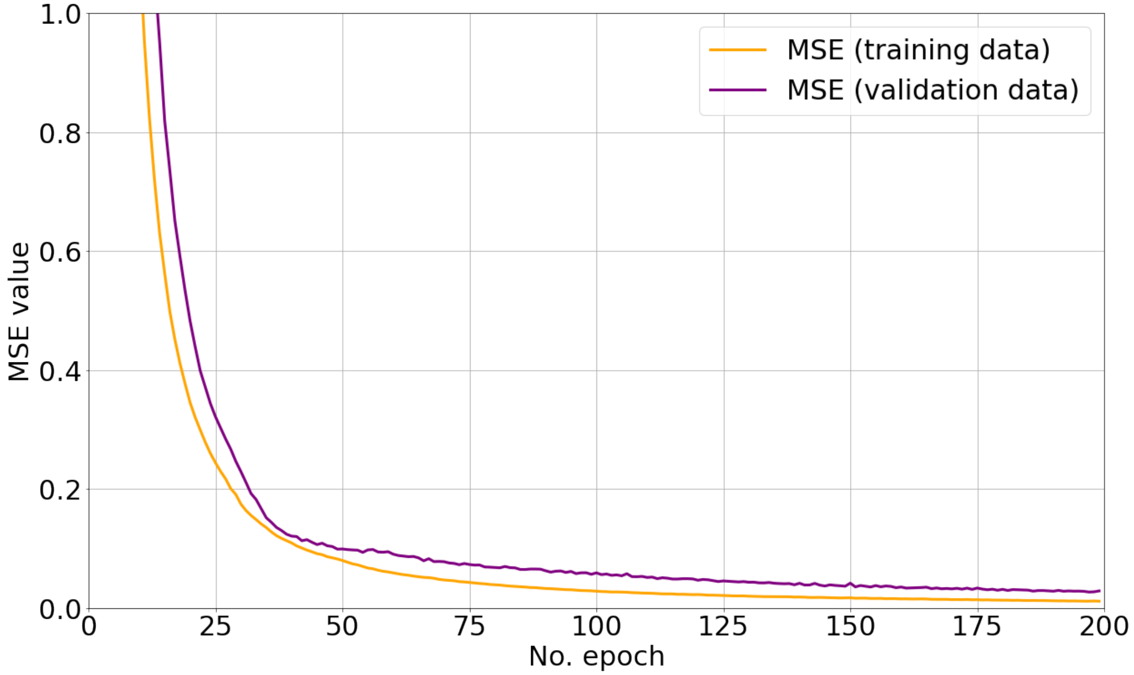


Figure 5.4: Training and evaluation error history over numbers of epochs for scenario I

Fig. 5.9 shows the test dataset prediction versus the ground truth of force magnitude, and for clarification, in Fig. 5.10 deviation from ground truth has been demonstrated. The MSE metrics for different liver geometries are plotted in Fig. 5.11.

As discussed in Chapter 4, principle component analysis and autoencoder were used for downsampling the input data. Fig. 5.12 shows the accuracy of the MSE error for various models based on PCA and AE.

To assess our proposed model's robustness, accuracy, and generalisability, four scenarios were considered based on synthetic data, where our model is trained, considering the variability of exertion point, hyperelastic properties, and liver geometries.

5.4 Scenario III : Variable Exertion Point, Identical Geometry, Variable Hyperelasticity

The following results of the model trained for a liver 3D model with various exertion points and variable hyperelasticity are shown. Table 5.5 shows the samples of Bayesian optimization and

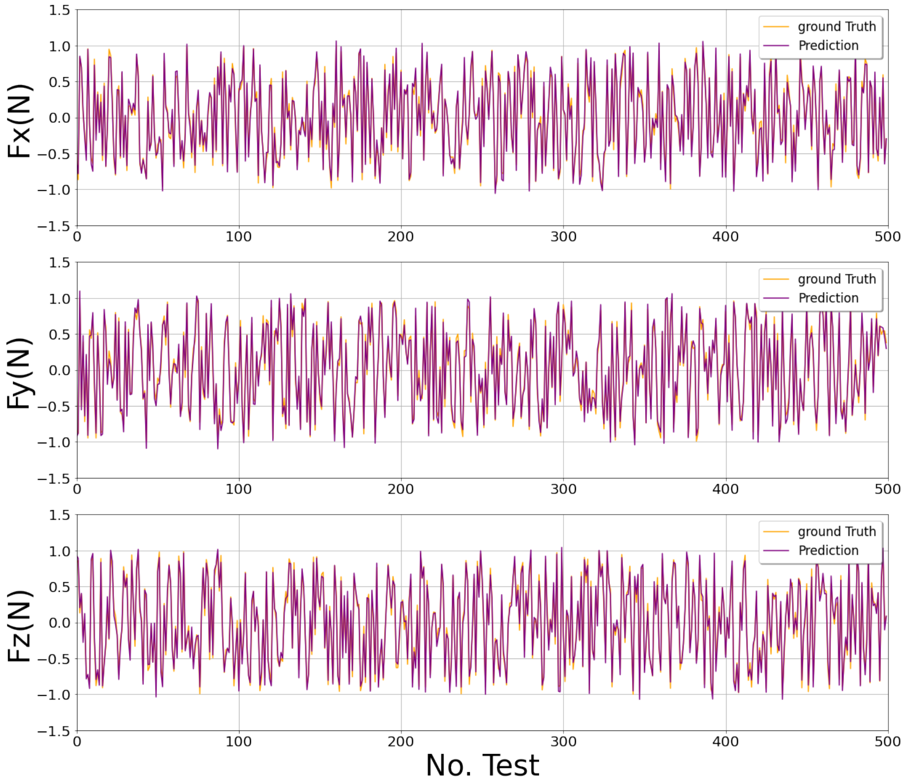


Figure 5.5: Test data force components prediction for scenario I

tunning of hyperparameters, where the best results are shown in table 5.6. Fig. 5.13 shows the Cv score based on 10 fold cross-validation of mean squared error function.

Fig. 5.14 confirms the convergence of training and evaluation error as the model is not over or underfitted. Same as previous section force prediction for 500 test set are depicted in Fig. 5.15 and for a better observation deviation of test dataset prediction versus ground truth are shown in Fig. 5.16.

5.5 Scenario IV : Variable Exertion point, Variable Geometry, Variable Hyperelasticity

Finally, A model that is trained on the dataset with various liver model and variable constitutive model and exertion coordinates. Tables 5.7 and 5.8 show the hyperparamter tunning and relative

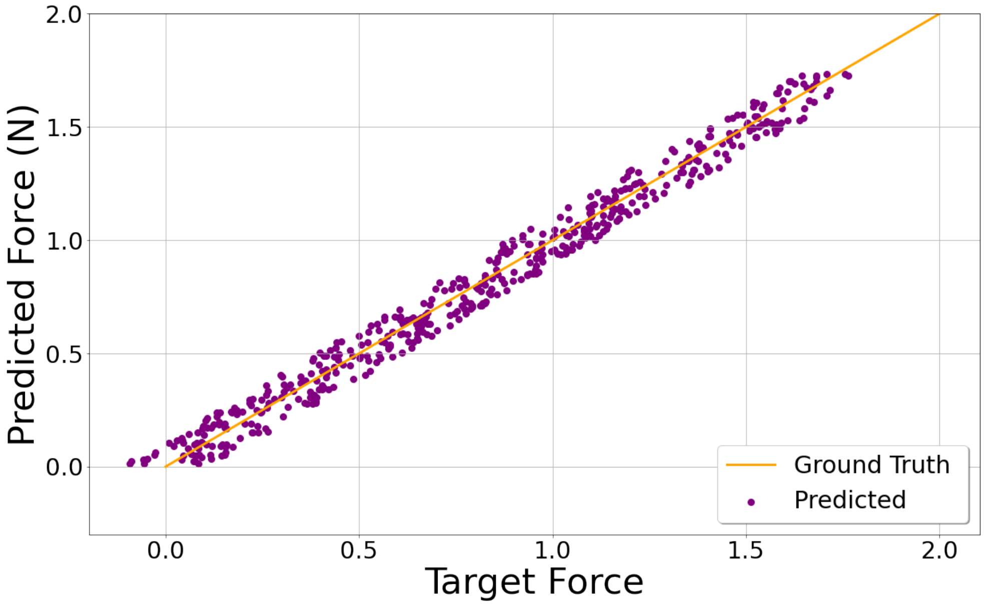


Figure 5.6: Randomly selected frame of force prediction vs. ground truth for scenario I

optimums. Fig. 5.17 shows the iteration versus CV score. Fig. 5.18 shows the training and evaluation error function convergence and force prediction over test dataset and randomly selected frames are shown in figures 5.19 and 5.20.

It is discernible that force prediction is independent of the direction of the force as shown in Figs. 5.5,5.9,5.15 and 5.19. Variational parameters diminish the accuracy as the model generalizes various conditions and features. However, the evaluation metrics are in an acceptable range. The force prediction's mean absolute error for training and test datasets is shown in Newton as well as the accuracy of the prediction with and without the presence of the proposed novel state vector in Fig. 5.21. The improvement is remarkable when the state vector is included, as illustrated in Table 5.9.

5.6 Validation: Encountering Unseen Liver Geometry and Hyperelasticity

To validate and assess the model's generalisability and accuracy, this study has tested the trained model on inputs obtained from an unseen liver geometry with unseen hyperelastic parameters.

Table 5.3: Hyperparameters tunning results of PYTORCH's AX optimization for second scenario

	$CVscore$	$Trial$	LR	DO	HL	HN	$Batchsize$
0	0.133008	0	0.000743	0.22896	1	290	64
23	0.189100	1	0.000146	0.0855886	2	267	256
115	0.071752	2	0.005245	0.26859	4	246	256
127	0.077281	3	0.008370	0.026772	2	118	64
139	0.084166	4	0.000771	0.030622	4	66	64
...	...	...	...	...	...	...	...
107	0.211679	195	0.008312	0.430908	1	145	8
108	0.091804	196	0.027259	0.022274	4	175	16
109	0.318355	197	0.041932	0.022399	4	150	8
110	0.090140	198	0.001035	0.026025	2	228	128
111	0.076729	199	0.000683	0.040717	3	214	8

LR = Learning rate
do = drop out
HL = Hidden Layer
HN = Hidden Neurons
BS= Batch size

Table 5.4: Hyperparameter with lowest CV score

Learning rate	0.0014602
Dropout rate	0.0152328
Hidden Layers	4
Neurons per layer	175
Batch size	64
Keras CV	0.067

Fig. 5.22 signals the predicted force compared to the unseen setting ground truth. The mechanical accuracy is over eighty-one percent.

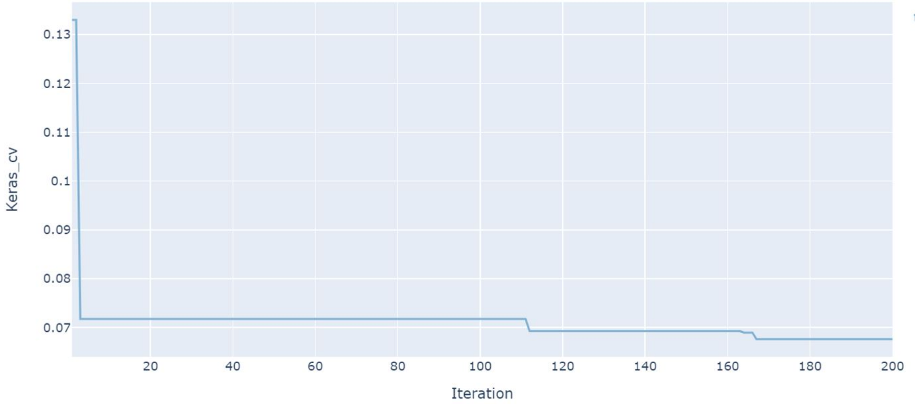


Figure 5.7: Hyperparameter tunning with AX Package powered by PYTORCH for scenario II

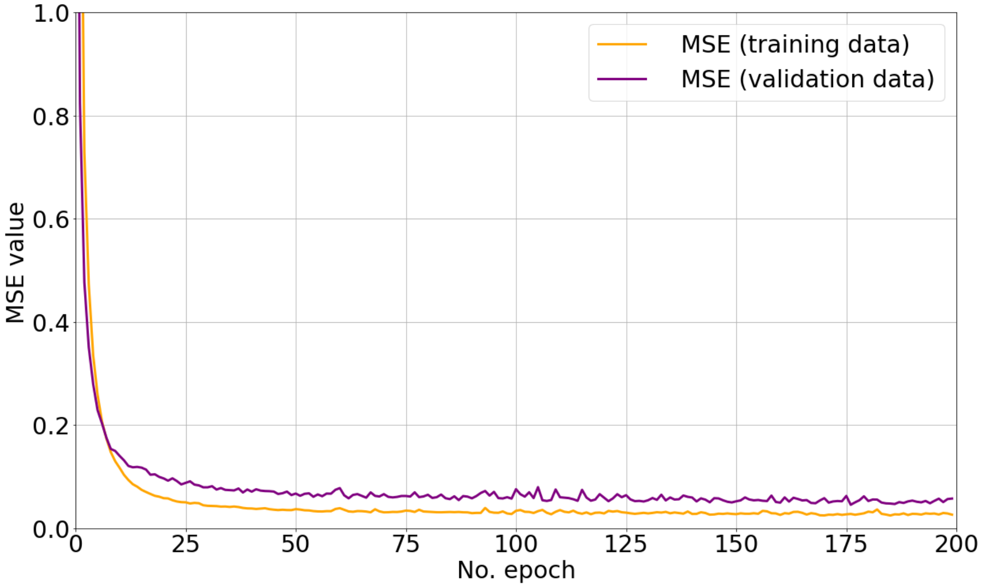


Figure 5.8: Training and evaluation error history over numbers of epochs for scenario II

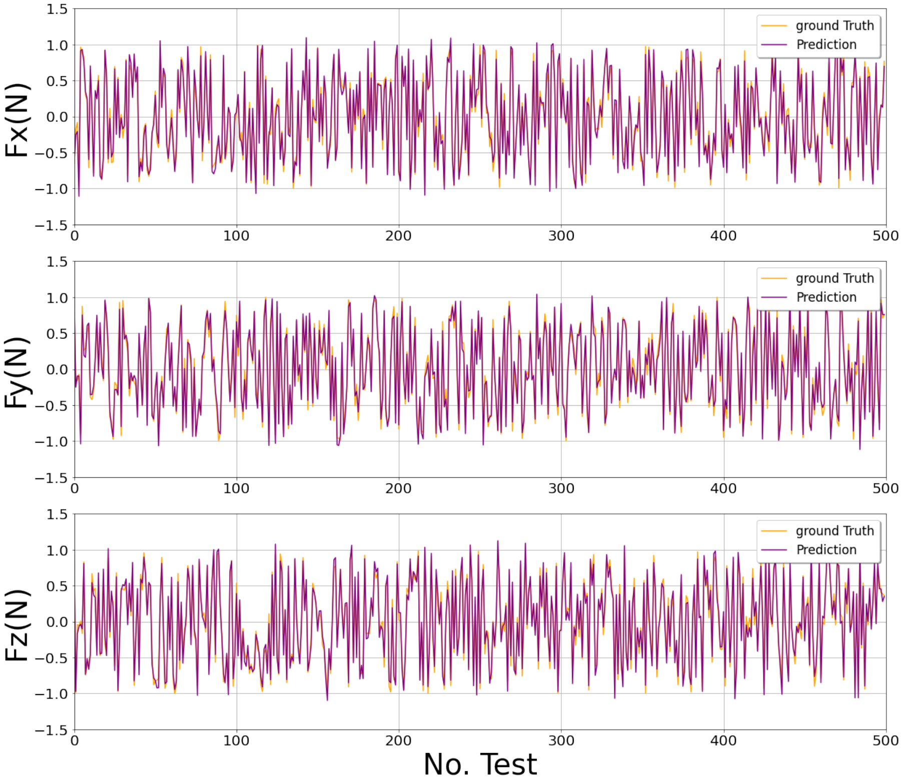


Figure 5.9: Test data force components prediction for scenario II

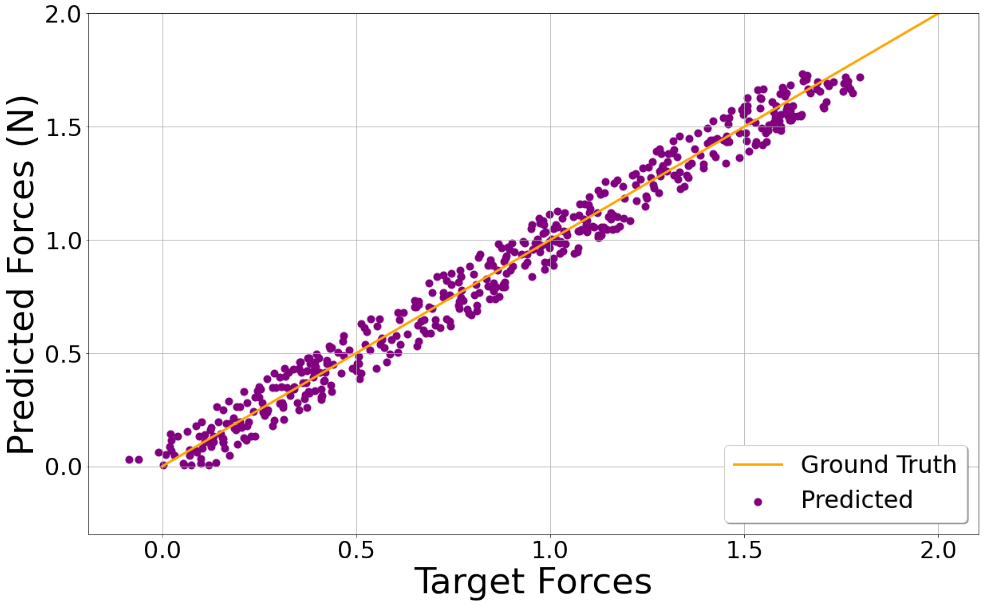


Figure 5.10: Randomly selected frame of force prediction vs. ground truth for a scenario II

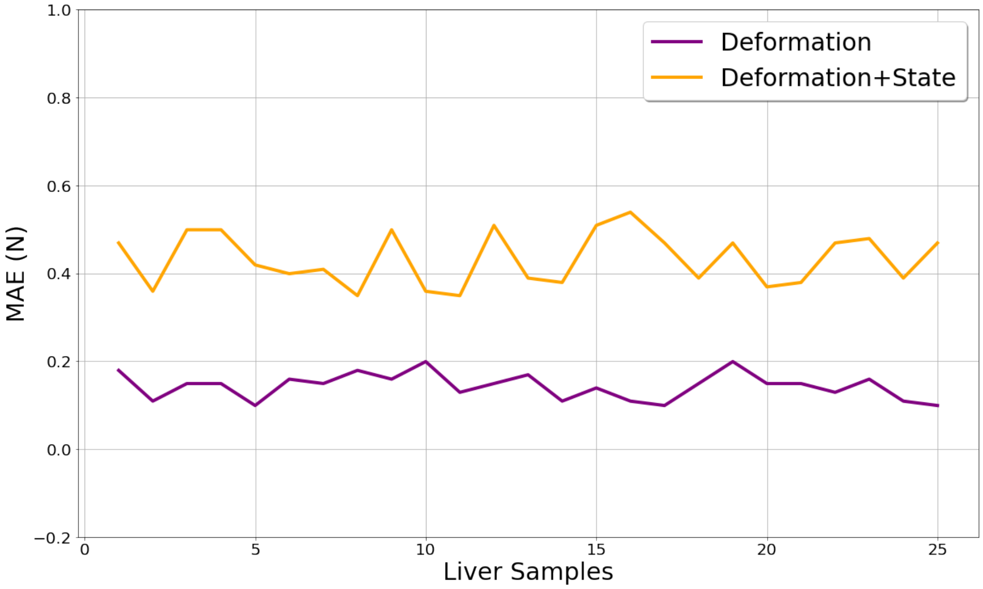


Figure 5.11: MSE for the model of scenario II trained on different liver geometries

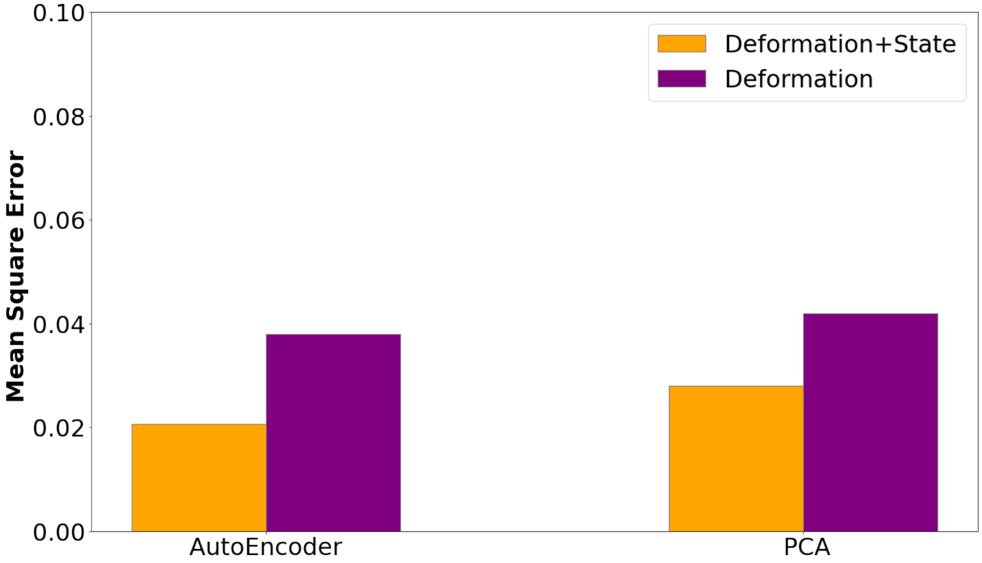


Figure 5.12: Principle component analysis vs autoencoder downsampling Impact in accuracy

Table 5.5: Hyperparameters optimization results with highest Cross Validation score

	$CV score$	$Trial$	LR	DO	HL	HN	$Batchsize$
0	0.076371	0	0.000188	0.035400	3	89	128
23	0.056859	1	0.001059	0.141951	3	241	16
115	0.055120	2	0.002885	0.025074	1	31	64
127	0.051696	3	0.047055	0.174349	1	295	32
139	0.095365	4	0.000235	0.010089	2	21	16
...	...	...	...	...	...	...	...
107	0.065477	195	0.002081	0.124934	4	45	8
108	0.333177	196	0.035932	0.069025	4	214	256
109	0.272884	197	0.002720	0.211302	4	37	256
110	0.107698	198	0.005425	0.143373	2	53	256
111	0.048810	199	0.003455	0.131242	1	207	128

LR = Learning rate
do = drop out
HL = Hidden Layer
HN = Hidden Neurons
BS= Batch size

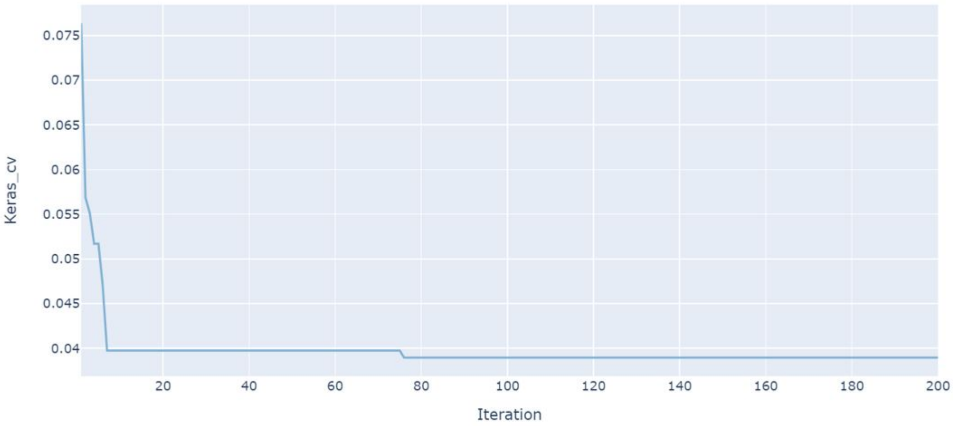


Figure 5.13: Hyperparameter tunning with AX package powered by PYTORCH for scenario III

Table 5.6: Hyperparameter with lowest CV score of scenario III

Learning rate	0.00023584
Dropout rate	0.013347
Hidden Layers	4
Neurons per layer	273
Batch size	16
Keras CV	0.038

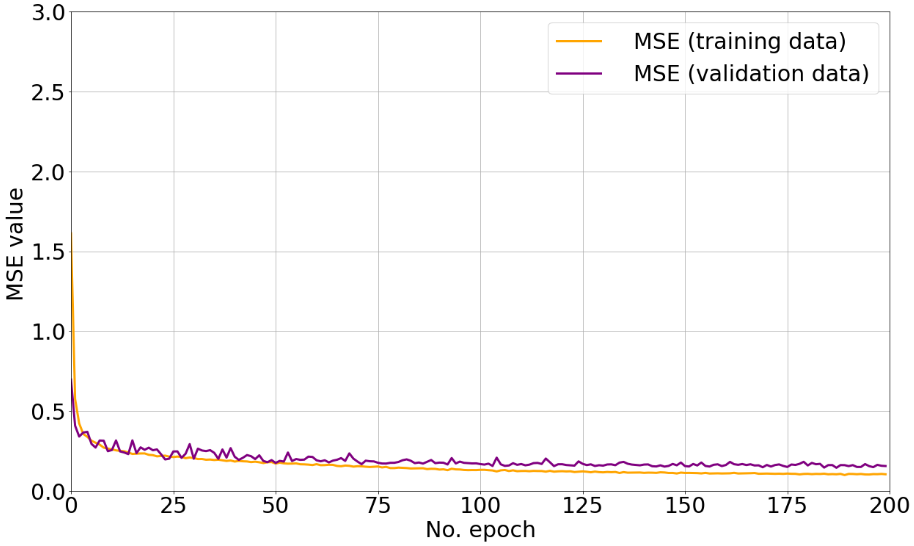


Figure 5.14: Training and evaluation error history over numbers of epochs for scenario III

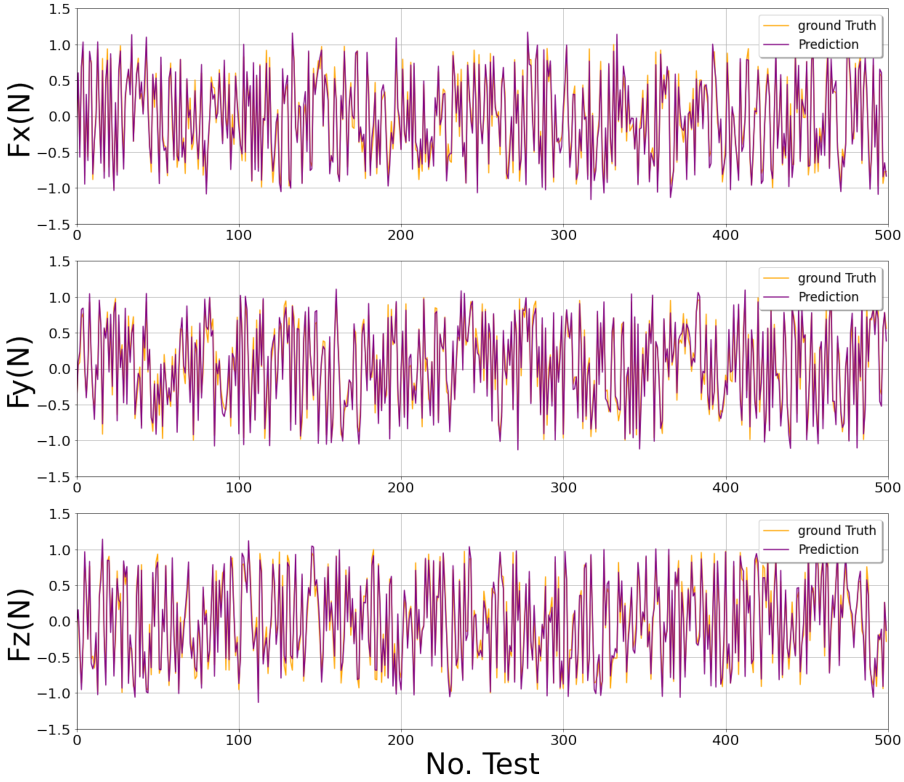


Figure 5.15: Test data force components prediction for scenario III

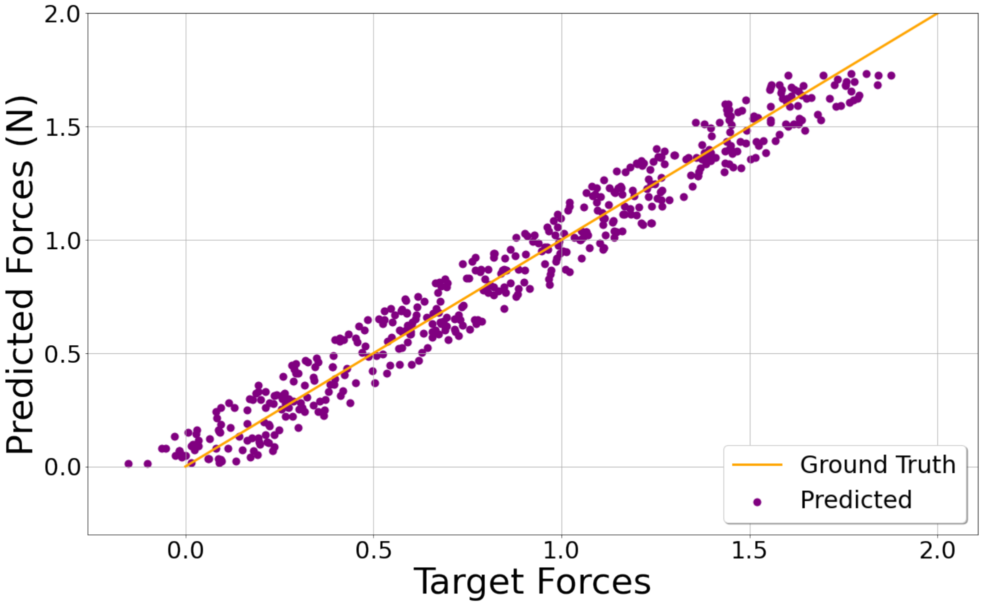


Figure 5.16: Randomly selected frame of force prediction vs. ground truth for a scenario III

Table 5.7: Hyperparameters optimization results with highest Cross Validation score

	$CV score$	$Trial$	LR	DO	HL	HN	$Batchsize$
0	0.057502	0	0.001466	0.088588	3	78	128
23	0.087833	1	0.001231	0.220350	2	195	64
115	0.920242	2	0.281379	0.431975	4	91	128
127	0.104570	3	0.008620	0.323838	3	277	64
139	2.812525	4	0.211940	0.261156	2	72	8
...	...	...	...	...	...	...	...
107	0.068487	195	0.000282	0.045984	1	186	64
108	0.338773	196	0.159757	0.318697	2	161	8
109	0.042253	197	0.000753	0.029664	3	297	256
110	0.333722	198	0.052195	0.072273	4	239	256
111	0.042190	199	0.000102	0.014943	4	289	128

LR = Learning rate
do = drop out
HL = Hidden Layer
HN = Hidden Neurons
BS= Batch size

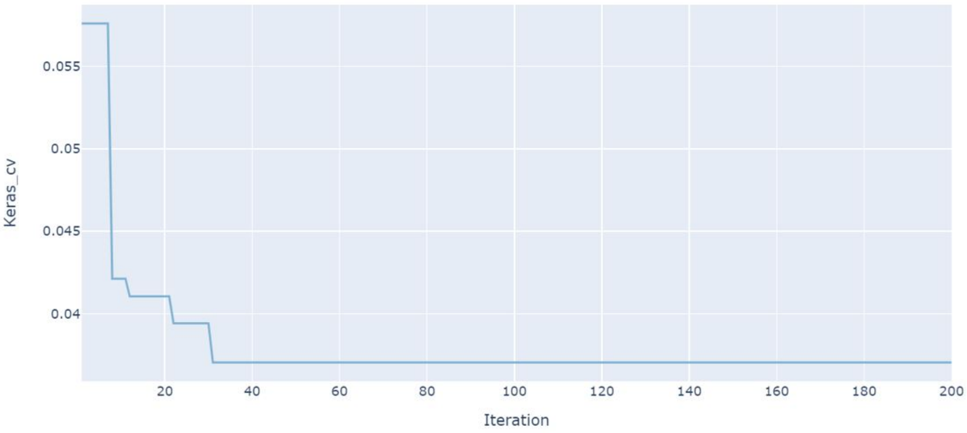


Figure 5.17: Hyperparameter tunning with AX Package powered by PYTORCH for scenario IV

Table 5.8: Hyperparameter with lowest CV score of scenario IV

Learning rate	0.024722
Dropout rate	0.020683
Hidden Layers	3
Neurons per layer	240
Batch size	256
Keras CV	0.0694

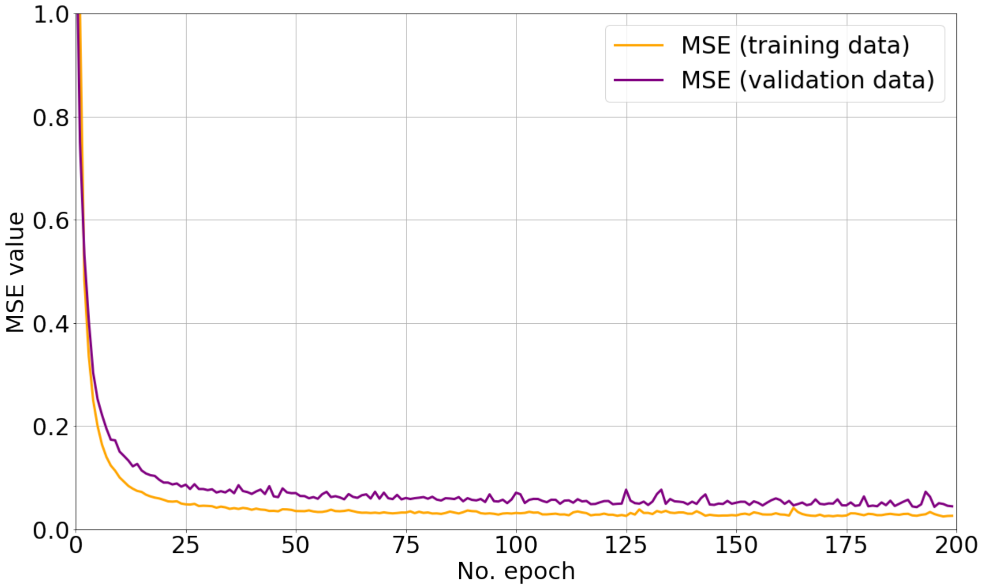


Figure 5.18: Training and evaluation error history over numbers of epochs for scenario IV

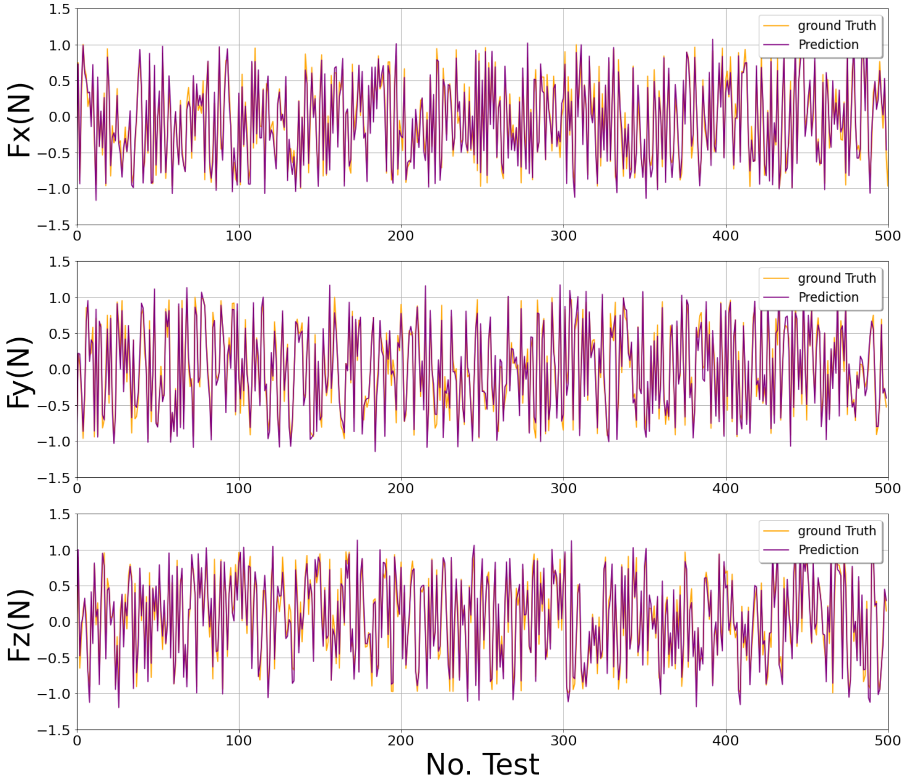


Figure 5.19: Test data force components prediction for scenario IV

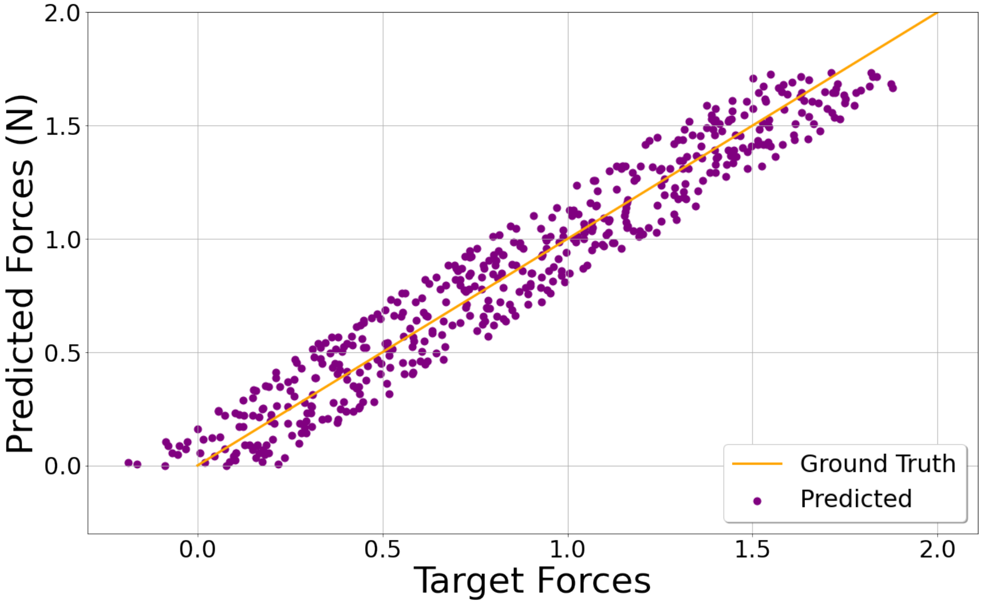


Figure 5.20: Randomly selected frame of force prediction vs. ground truth for a scenario IV

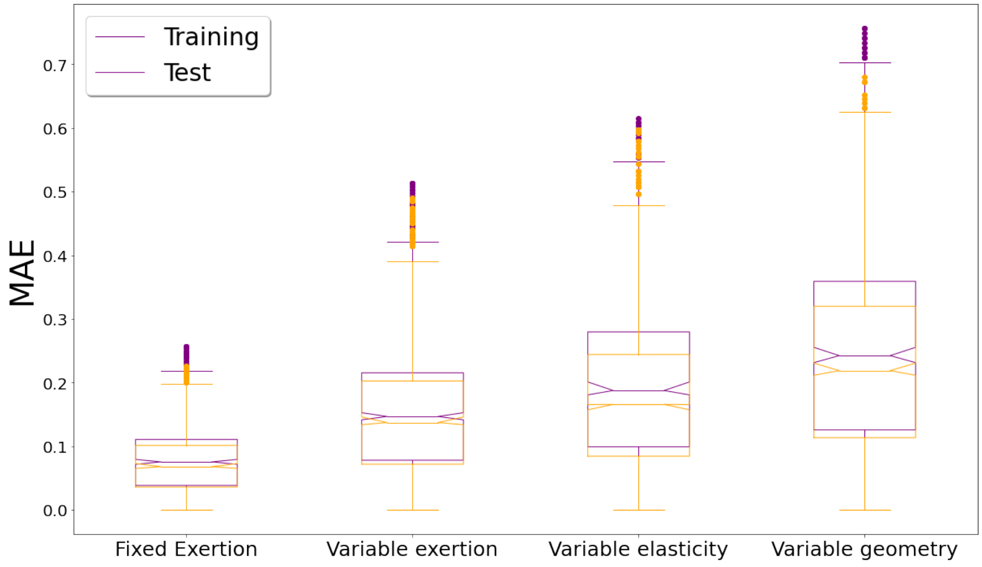


Figure 5.21: Absolute error of force prediction for different scenarios

Table 5.9: Prediction accuracy for different scenarios

Method	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Deformation	92.12	78.14	69.11	65.66
Deformation + State	95.76	91.22	88.53	84.81

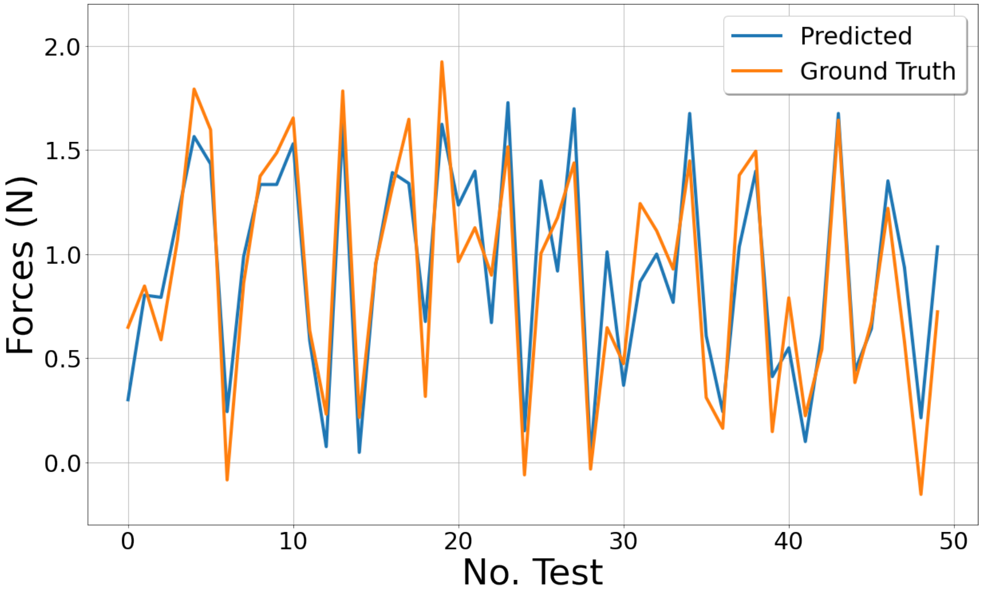


Figure 5.22: Absolute error of force prediction for different scenarios

Chapter 6

Conclusion

This thesis presented a study to predict the force exerted on the liver soft tissue based on visual information of surface deformation along with preoperative information of the organ. Unlike most previous research where data are collected and labeled experimentally, the organ deformation was simulated, considering actual boundary conditions and the biomechanical characteristics to provide the hyperelastic deformation synthetic dataset. The patient-specific liver digital twins have been reconstructed from CT images, and various hyperelastic models were assessed for various exertion positions concerning liver geometrical factors. Twenty-four thousand simulations have been executed and automatized to provide a sufficient amount of labeled data for the training purpose of a deep neural network. Considering the scale of our simulation number, retrieving and restoring the simulation is highly time-consuming and not manually practical. Therefore, ANSYS parametric design language (APDL), a powerful scripting language, has been used to customize simulation features to automatize the process of simulation, retrieving, and restoring the results, which reduces the time consumed remarkably and enables us to create as much data as needed. Simulations were used to train the machine to predict the force that caused the specific deformation of the geometry.

Our research shows that a limited region of surface observation neighboring the position of exertion, named region of interest (ROI), is adequate for our prediction. Knowing the prior information on the liver model, it was concluded that approximately less than 20 percent of the total surface would produce a good result. Hence, all our prediction has been performed based on an 18 percent ROI. Our model has been trained based on the point cloud of the surface deformation of ROI. In practice, this point cloud can be acquired from the patient's in-vivo stereoscopic image of MIS with 3D map reconstruction techniques that have been widely addressed in the literature. Thanks to recent improvements in image processing and computer vision techniques mentioned, 3D geometry can be acquired in real-time.

One of the main drawbacks of image-based convolutional networks is the impact of stereo camera and lighting position in the image that may cause perturbation and error in force prediction. By using the 3D point cloud as the input in our model, this issue would be prevented significantly.

As mentioned previously, proposed models mainly deployed labeled data from exvivo settings since measuring forces in an invivo setting and labeling the data is not technically feasible. Hence the acquired data in experiments are generally limited due to the simplified models in terms of geometry, biomechanical parameters, and the actual behavior of the tissue in operating conditions. Our model generalizes these features since it is viable to change mentioned parameters in the simulation easily. In contrast, in experiments, these changes are relatively costly. In the context of machine learning, it is almost impossible to diversify the experimental data with respect to these features to give the machine the capability to distinguish the differences that these features may impose on the final results. In addition, Unlike the experiment limitation, our model considers the organ boundary interaction with other organs. In other words, experiments are mainly implemented on a well-defined geometry of an artificial tissue or a piece of a cadaver. The organ biomechanics and the impact of other organs' contact are not considered; therefore, the in-vivo applicability of these models is pretty questionable. Additionally, a denoising model was presented, named denoising autoencoder, to diminish the uncertainty caused due to tool's occlusion in the MIS region.

Not only does our model consider various patient-specific liver models, but also various exertion positions can be assessed. Results have been illustrated in chapter 5 for four scenarios in each, and the machine was trained based on the provided dataset and state vector of preoperative information. Since the presence of the surgical tool in the image would deteriorate the 3D reconstruction and act as noise, this model has addressed that issue by eliminating a portion of nodes surrounding the exertion point to consider this perturbation as if we do not have access to this region. Although this assumption reduces the accuracy, it is still higher than the previous method, and it is worth mentioning that this result can be obtained for an invivo setting.

6.1 Future Works

This study can be continued in different aspects. From enhancing the patient-based liver models datasets to include more information about various geometries, especially with the presence of tumors. Since genuinely biomechanical specifications of tumors vary from the liver tissue, this matter would impact the overall behavior of the organ encountering an external force.

Furthermore, Through transfer learning, the trained model on the FEA dataset can be linked to actual surgery data to improve the compatibility of the proposed model. Undoubtedly the same approach can be deployed for any other organ with a reliable and realistic dataset derived from an accurate simulation.

Bibliography

[1] A. M. Okamura, “Haptic feedback in robot-assisted minimally invasive surgery,” *Current opinion in urology*, vol. 19, no. 1, p. 102, 2009.

[2] Y.-J. Lim, D. Deo, T. P. Singh, D. B. Jones, and S. De, “In situ measurement and modeling of biomechanical response of human cadaveric soft tissues for physics-based surgical simulation,” *Surgical endoscopy*, vol. 23, no. 6, pp. 1298–1307, 2009.

[3] B. K. Tay, J. Kim, and M. A. Srinivasan, “In vivo mechanical behavior of intra-abdominal organs,” *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 11, pp. 2129–2138, 2006.

[4] T. Haidegger, B. Benyó, L. Kovács, and Z. Benyó, “Force sensing and force control for surgical robots,” *IFAC Proceedings Volumes*, vol. 42, no. 12, pp. 401–406, 2009.

[5] T. Ritto and F. Rochinha, “Digital twin, physics-based model, and machine learning applied to damage detection in structures,” *Mechanical Systems and Signal Processing*, vol. 155, p. 107614, 2021.

[6] P. Seventekidis and D. Giagopoulos, “A combined finite element and hierarchical deep learning approach for structural health monitoring: Test on a pin-joint composite truss structure,” *Mechanical Systems and Signal Processing*, vol. 157, p. 107735, 2021.

[7] P. Seventekidis, D. Giagopoulos, A. Arailopoulos, and O. Markogiannaki, “Structural health monitoring using deep learning with optimal finite element model generated data,” *Mechanical Systems and Signal Processing*, vol. 145, p. 106972, 2020.

[8] A. Fernandez-Navamuel, D. Zamora-Sánchez, Á. J. Omella, D. Pardo, D. Garcia-Sanchez, and

F. Magalhães, "Supervised deep learning with finite element simulations for damage identification in bridges," *Engineering Structures*, vol. 257, p. 114016, 2022.

[9] E. P. Westebring-van der Putten, R. H. Goossens, J. J. Jakimowicz, and J. Dankelman, "Haptics in minimally invasive surgery–a review," *Minimally Invasive Therapy & Allied Technologies*, vol. 17, no. 1, pp. 3–16, 2008.

[10] M. Tavakoli, R. V. Patel, M. Moallen, and A. Aziminejad, *Haptics for teleoperated surgical robotic systems*. World Scientific, 2008, vol. 1.

[11] A. M. Okamura, "Methods for haptic feedback in teleoperated robot-assisted surgery," *Industrial Robot: An International Journal*, 2004.

[12] M. A. Qasaimeh, S. Sokhanvar, J. Dargahi, and M. Kahrizi, "Pvdf-based microfabricated tactile sensor for minimally invasive surgery," *Journal of Microelectromechanical Systems*, vol. 18, no. 1, pp. 195–207, 2008.

[13] R. Ahmadi, M. Packirisamy, J. Dargahi, and R. Cecere, "Discretely loaded beam-type optical fiber tactile sensor for tissue manipulation and palpation in minimally invasive robotic surgery," *IEEE Sensors Journal*, vol. 12, no. 1, pp. 22–32, 2011.

[14] N. Bandari, J. Dargahi, and M. Packirisamy, "Tactile sensors for minimally invasive surgery: a review of the state-of-the-art, applications, and perspectives," *Ieee Access*, vol. 8, pp. 7682–7708, 2019.

[15] N. Zemiti, G. Morel, T. Ortmaier, and N. Bonnet, "Mechatronic design of a new robot for force control in minimally invasive surgery," *IEEE/ASME Transactions On Mechatronics*, vol. 12, no. 2, pp. 143–153, 2007.

[16] J. Rosen, J. D. Brown, L. Chang, M. Barreca, M. Sinanan, and B. Hannaford, "The bluedragon-a system for measuring the kinematics and dynamics of minimally invasive surgical tools in-vivo," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2. IEEE, 2002, pp. 1876–1881.

[17] B. Demi, T. Ortmaier, and U. Seibold, “The touch and feel in minimally invasive surgery,” in *IEEE international workshop on haptic audio visual environments and their applications*. IEEE, 2005, pp. 6–pp.

[18] E. U. Braun, H. Mayer, A. Knoll, R. Lange, and R. Bauernschmitt, “The must-have in robotic heart surgery: haptic feedback,” *Medical robotics*, pp. 9–20, 2008.

[19] A. H. H. Hosseinabadi, M. Honarvar, and S. E. Salcudean, “Optical force sensing in minimally invasive robotic surgery,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4033–4039.

[20] A. C. Goh, D. W. Goldfarb, J. C. Sander, B. J. Miles, and B. J. Dunkin, “Global evaluative assessment of robotic skills: validation of a clinical assessment tool to measure robotic surgical skills,” *The Journal of urology*, vol. 187, no. 1, pp. 247–252, 2012.

[21] R. Sánchez, O. Rodríguez, J. Rosciano, L. Vegas, V. Bond, A. Rojas, and A. Sanchez-Ismayel, “Robotic surgery training: construct validity of global evaluative assessment of robotic skills (gears),” *Journal of robotic surgery*, vol. 10, no. 3, pp. 227–231, 2016.

[22] A. J. Hung, J. Chen, and I. S. Gill, “Automated performance metrics and machine learning algorithms to measure surgeon performance and anticipate clinical outcomes in robotic surgery,” *JAMA surgery*, vol. 153, no. 8, pp. 770–771, 2018.

[23] F. Piqué, M. N. Boushaki, M. Brancadoro, E. De Momi, and A. Menciassi, “Dynamic modeling of the da vinci research kit arm for the estimation of interaction wrench,” in *2019 International Symposium on Medical Robotics (ISMR)*. IEEE, 2019, pp. 1–7.

[24] M. Wang and Y. Zeng, “Asking the right questions to elicit product requirements,” *International Journal of Computer Integrated Manufacturing*, vol. 22, no. 4, pp. 283–298, Apr. 2009. [Online]. Available: http://dx.doi.org/10.1080/09511920802232902

[25] G. A. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano, “Modelling and identification of the da vinci research kit robotic arms,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1464–1469.

[26] A. Gao, N. Liu, M. Shen, M. EMK Abdelaziz, B. Temelkuran, and G.-Z. Yang, “Laser-profiled continuum robot with integrated tension sensing for simultaneous shape and tip force estimation,” *Soft Robotics*, vol. 7, no. 4, pp. 421–443, 2020.

[27] Y. Zhou, B. J. Nelson, and B. Vikramaditya, “Integrating optical force sensing with visual servoing for microassembly,” *Journal of Intelligent and Robotic Systems*, vol. 28, no. 3, pp. 259–276, 2000.

[28] L. Soler, S. Nicolau, P. Pessaux, D. Mutter, and J. Marescaux, “Real-time 3d image reconstruction guidance in liver resection surgery,” *Hepatobiliary surgery and nutrition*, vol. 3, no. 2, p. 73, 2014.

[29] M. T. Thai, P. T. Phan, T. T. Hoang, S. Wong, N. H. Lovell, and T. N. Do, “Advanced intelligent systems for surgical robotics,” *Advanced Intelligent Systems*, vol. 2, no. 8, p. 1900138, 2020.

[30] C. W. Kennedy and J. P. Desai, “A vision-based approach for estimating contact forces: Applications to robot-assisted surgery,” *Applied Bionics and Biomechanics*, vol. 2, no. 1, pp. 53–60, 2005.

[31] W. Kim, S. Seung, H. Choi, S. Park, S. Y. Ko, and J.-O. Park, “Image-based force estimation of deformable tissue using depth map for single-port surgical robot,” in *2012 12th International Conference on Control, Automation and Systems*. IEEE, 2012, pp. 1716–1719.

[32] E. Noohi, S. Parastegari, and M. Žefran, “Using monocular images to estimate interaction forces during minimally invasive surgery,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 4297–4302.

[33] S. Giannarou, M. Ye, G. Gras, K. Leibrandt, H. J. Marcus, and G.-Z. Yang, “Vision-based deformation recovery for intraoperative force estimation of tool–tissue interaction for neurosurgery,” *International journal of computer assisted radiology and surgery*, vol. 11, no. 6, pp. 929–936, 2016.

[34] N. Haouchine, W. Kuang, S. Cotin, and M. Yip, “Vision-based force feedback estimation for

robot-assisted surgery using instrument-constrained biomechanical three-dimensional maps," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2160–2165, 2018.

[35] M. A. Greminger and B. J. Nelson, "Modeling elastic objects with neural networks for vision-based force measurement," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 2. IEEE, 2003, pp. 1278–1283.

[36] A. I. Aviles, A. Marban, P. Sobrevilla, J. Fernandez, and A. Casals, "A recurrent neural network approach for 3d vision-based force estimation," in *2014 4th International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2014, pp. 1–6.

[37] A. I. Aviles, S. Alsaleh, P. Sobrevilla, and A. Casals, "Sensorless force estimation using a neuro-vision-based approach for robotic-assisted surgery," in *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE, 2015, pp. 86–89.

[38] A. I. Aviles, S. M. Alsaleh, E. Montseny, P. Sobrevilla, and A. Casals, "A deep-neuro-fuzzy approach for estimating the interaction forces in robotic surgery," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2016, pp. 1113–1119.

[39] M. C. Yip, S. G. Yuen, and R. D. Howe, "A robust uniaxial force sensor for minimally invasive surgery," *IEEE transactions on biomedical engineering*, vol. 57, no. 5, pp. 1008–1011, 2010.

[40] A. Faragasso, J. Bimbo, Y. Noh, A. Jiang, S. Sareh, H. Liu, T. Nanayakkara, H. A. Wurdemann, and K. Althoefer, "Novel uniaxial force sensor based on visual information for minimally invasive surgery," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1405–1410.

[41] A. I. Aviles, S. M. Alsaleh, J. K. Hahn, and A. Casals, "Towards retrieving force feedback in robotic-assisted surgery: A supervised neuro-recurrent-vision approach," *IEEE transactions on haptics*, vol. 10, no. 3, pp. 431–443, 2016.

[42] F. Baghaei Naeini, D. Makris, D. Gan, and Y. Zweiri, "Dynamic-vision-based force measurements using convolutional recurrent neural networks," *Sensors*, vol. 20, no. 16, p. 4469, 2020.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[44] N. Gessert, J. Beringhoff, C. Otte, and A. Schlaefer, "Force estimation from oct volumes using 3d cnns," *International journal of computer assisted radiology and surgery*, vol. 13, no. 7, pp. 1073–1082, 2018.

[45] Z. Chua, A. M. Jarc, and A. M. Okamura, "Toward force estimation in robot-assisted surgery using deep learning with vision and robot state," *arXiv preprint arXiv:2011.02112*, 2020.

[46] A. Marban, V. Srinivasan, W. Samek, J. Fernández, and A. Casals, "Estimation of interaction forces in robotic surgery using a semi-supervised deep neural network model," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 761–768.

[47] ——, "A recurrent convolutional neural network approach for sensorless force estimation in robotic surgery," *Biomedical Signal Processing and Control*, vol. 50, pp. 134–150, 2019.

[48] A. A. Nazari, F. Janabi-Sharifi, and K. Zareinia, "Image-based force estimation in medical applications: A review," *IEEE Sensors Journal*, 2021.

[49] A. R. Kemper, A. C. Santago, J. D. Stitzel, J. L. Sparks, and S. M. Duma, "Biomechanical response of human liver in tensile loading," in *Annals of advances in automotive medicine/annual scientific conference*, vol. 54. Association for the Advancement of Automotive Medicine, 2010, p. 15.

[50] I. Ansys, "Ansys mechanical apdl element reference," *ANSYS, Inc., USA*, 2011.

[51] C. L. Nota, I. H. B. Rinkes, I. Q. Molenaar, H. C. van Santvoort, Y. Fong, and J. Hagendoorn, "Robot-assisted laparoscopic liver resection: a systematic review and pooled analysis of minor and major hepatectomies," *Hpb*, vol. 18, no. 2, pp. 113–120, 2016.

[52] J. Yang, L. Yu, L. Wang, H. Li, and Q. An, "Study on mechanical characterization of liver tissue based on haptic devices for virtual surgical simulation," *Journal of Mechanics in Medicine and Biology*, vol. 16, no. 08, p. 1640016, 2016.

[53] H. Delingette and N. Ayache, "Soft tissue modeling for surgery simulation," *Handbook of Numerical Analysis*, vol. 12, pp. 453–550, 2004.

[54] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J.-C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka *et al.*, "3d slicer as an image computing platform for the quantitative imaging network," *Magnetic resonance imaging*, vol. 30, no. 9, pp. 1323–1341, 2012.

[55] J. Bonet and R. D. Wood, *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press, 1997.

[56] Y. Li and D. X. Dai, *Biomechanical engineering of textiles and clothing*. Woodhead Publishing, 2006.

[57] Z. Gao, K. Lister, and J. P. Desai, "Constitutive modeling of liver tissue: experiment and theory," *Annals of biomedical engineering*, vol. 38, no. 2, pp. 505–516, 2010.

[58] E. Roan and K. Vemaganti, "The nonlinear material properties of liver tissue determined from no-slip uniaxial compression experiments," *Journal of biomechanical engineering*, vol. 129, no. 3, pp. 450–456, 2007.

[59] S. Nikolaev and S. Cotin, "Estimation of boundary conditions for patient-specific liver simulation during augmented surgery," *International Journal of Computer Assisted Radiology and Surgery*, vol. 15, pp. 1107–1115, 2020.

[60] "ANSYS engineering simulation software," https://www.ansys.com/.

[61] F. G. Hamza-Lup, A. Seitan, D. M. Popovici, and C. M. Bogdan, "Medical simulation and training:" haptic" liver," *arXiv preprint arXiv:1812.03325*, 2018.

[62] G. Zampokas, K. Tsiolis, G. Peleka, I. Mariolis, S. Malasiotis, and D. Tzovaras, “Real-time 3d reconstruction in minimally invasive surgery with quasi-dense matching,” in *2018 IEEE International Conference on Imaging Systems and Techniques (IST)*. IEEE, 2018, pp. 1–6.

[63] A. Roberti, N. Piccinelli, F. Falezza, G. De Rossi, S. Bonora, F. Setti, P. Fiorini, and R. Muradore, “A time-of-flight stereoscopic endoscope for anatomical 3d reconstruction,” in *2021 International Symposium on Medical Robotics (ISMR)*. IEEE, 2021, pp. 1–7.

[64] I. T. Jolliffe and J. Cadima, “Principal component analysis: a review and recent developments,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016.

[65] “PYTORCH adaptive experimentation platform,” https://ax.dev/.

[66] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1, no. 2.

[67] J. Zhang, Y. Zhong, and C. Gu, “Neural network modelling of soft tissue deformation for surgical simulation,” *Artificial intelligence in medicine*, vol. 97, pp. 61–70, 2019.