Evaluation of U-Net model in the detection of cervical spine fractures

by

Faranak Kheirandish

A thesis submitted in partial fulfilment
of the requirements for the degree of
Master of Science (MSc) in Computational Science

The Office of Graduate Studies
Laurentian University
Sudbury, Ontario, Canada

©Faranak, 2023

THESIS DEFENCE COMMITTEE/COMITÉ DE SOUTENANCE DE THÈSE
Laurentian Université/Université Laurentienne
Office of Graduate Studies/Bureau des études supérieures

Title of Thesis Titre de la thèse	Evaluation of U-Net model in the detection of cervical spine fractures	
Name of Candidate Nom du candidat	Kheirandish, Faranak	
Degree Diplôme	Master of Science	
Department/Program Département/Programme	Computational Sciences	Date of Defence Date de la soutenance October 18, 2023

APPROVED/APPROUVÉ

Thesis Examiners/Examinateurs de thèse:

Dr. Amr Abdel-Dayem
(Supervisor/Directeur(trice) de thèse)

Dr. Meysar Zeinali-Ghayeshghorshagh

Dr. Meng Cheng Lau
(Committee member/Membre du comité)

Dr. Sabah Mohammed
(External Examiner/Examinateur externe)

Approved for the Office of Graduate Studies
Approuvé pour le Bureau des études supérieures
Tammy Eger, PhD
Vice-President Research (Office of Graduate Studies)
Vice-rectrice à la recherche (Bureau des études supérieures)
Laurentian University / Université Laurentienne

ACCESSIBILITY CLAUSE AND PERMISSION TO USE

I, **Faranak Kheirandish**, hereby grant to Laurentian University and/or its agents the non-exclusive license to archive and make accessible my thesis, dissertation, or project report in whole or in part in all forms of media, now or for the duration of my copyright ownership. I retain all other ownership rights to the copyright of the thesis, dissertation or project report. I also reserve the right to use in future works (such as articles or books) all or part of this thesis, dissertation, or project report. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that this copy is being made available in this form by the authority of the copyright owner solely for the purpose of private study and research and may not be copied or reproduced except as permitted by the copyright laws without written authority from the copyright owner.

Abstract

The cervical spine is composed of seven vertebrae from C1 to C7 with a lordotic curve (C-shaped curve) and joints between vertebrae for spine mobility. A computed tomography (CT) is commonly used by experts and physicians in imaging diagnosis to give information about the cervical spine and vertebrae in the neck. Diseases such as spinal stenosis (narrowing of the spinal canal), herniated discs, tumors, and fractures in the cervical spine can be diagnosed by CT scans. Quickly detecting the presence, and location of cervical spine fractures in CT scans helps physicians prevent neurologic deterioration and paralysis after trauma. Throughout this thesis, a U-Net model was trained for semantic segmentation on approximately 2019 study instances with provided CT images, while only 87 of them have been segmented by spine radiology specialists. After that, a combination of 2D CNN and bidirectional GRU deep learning models was used for the detection of fractures in each vertebra, as a classification task.

The objectives of this research are to develop two deep-learning models for detecting and localizing cervical spine fractures and evaluate the ongoing research activities on semantic segmentation and classification in the medical field. This research aims to use a semantic segmentation algorithm in deep learning by using U-Net architecture to estimate the location of each cervical vertebra, as well as propose a deep convolutional neural network (DCNN) with a bidirectional GRU memory (Bi-GRU) layer for the automated detection of cervical spine fractures in CT images. This approach was trained and tested on a dataset provided by RSNA (a team of the American Society of Neuroradiology and Spine Radiology).

Furthermore, the critical factors, such as preprocessing techniques and specialized loss functions

were explored that must be taken into consideration when segmenting 3D medical images. Whether used as a standalone framework for segmentation and classification tasks or as an integrated backbone for medical image processing, this architecture is flexible enough to accommodate other models. The proposed approach yields results that are comparable to those of existing techniques, but it can be improved by using larger image sizes and more advanced GPU workstations that will reduce the overall processing time.

Future research will be using other pretrained networks as an encoder and increasing image sizes to examin the performance improvmet of the architecture which needed more advanced computational resources and also integrate the current architecture into a simulated crash scenarios to use in various applications such as producing protecive sport equipments.

Dedication

This thesis is dedicated to my family, especially, my husband Omid Soltani, and my parents Shahla Khazra and Mohammad Hossein Kheirandish, who support me in this program.

Acknowledgments

I would like to express my appreciation to Laurentian University through the Bharti school of engineering and computer science, which gave me the opportunity to pursue a master's degree and supported the research that I developed.

I am extremely grateful to my supervisor, Dr. Amr Abdel Dayem, who has provided advice and intellectual contribution to my research in a wonderful environment and motivation. Without his continuous discussions and encouragement, it could be extremely difficult to get through my dissertation.

Contents

Abstract **iii**

Dedication **v**

Acknowledgements **vi**

List of Figures **x**

List of Tables **xiii**

List of Abbreviations **xiv**

1 Introduction **1**

1.1 problem statement and motivations 1

1.2 Objectives 2

1.3 Outline of this Thesis 3

2 Theory and Background Materials **4**

2.1 Cervical spine anatomy 4

2.2 Cervical spine injuries and disorders 9

2.3 Cervical Spine fracture detection 11

2.3.1 Diagnostic Techniques: 11

2.3.2 Image segmentation 13

2.4 Deep Learning and its architectures . . . 15
2.4.1 Deep learning . . . 15
2.4.2 Semantic segmentation and U-Net . . . 20
2.4.3 CNN with Bidirectional GRU layer . . . 23
3 Literature review **26**
3.1 Previous approaches . . . 26
4 Modelling and DataSet **33**
4.1 Exploratory Data Analysis . . . 33
4.1.1 Dataset analysis . . . 33
4.1.2 Image Format . . . 35
4.2 Model training . . . 39
4.2.1 3D Semantic Segmentation model . . . 39
4.2.2 Fracture detection model . . . 44
5 3D Segmentation model **48**
5.1 3D U-Net implementation and hyperparameter tunning . . . 48
5.2 Experimental Results . . . 51
5.3 Discussion . . . 57
6 Classification model **59**
6.1 Yolov5 implementation . . . 59
6.2 classification . . . 63
6.2.1 Processing the training data . . . 63
6.2.2 Implementation of data augmentation . . . 64
6.2.3 CNN + GRU Implementation . . . 66
6.3 Experimental Results . . . 70
6.4 Discussion . . . 72

7 Concluding remarks and future works **74**

List of Figures

2.1 the anatomy of skull and cervical spine from greatbigcanvas [27] 5
2.2 the structure of the cervical spine, the uppermost portion of the spine from Musculoskeletal [61] 6
2.3 A superior view of one typical vertebra from Peter et al. [9] 7
2.4 intervertebral view from the American Academy of Orthopaedic Surgeons [14] . . 7
2.5 Spinal Ligaments from Health Central [65] 8
2.6 Spinal Cord injury from INSYNC PHISIO [52] 9
2.7 Architecture of U-Net based on the paper by Ronneberger et.al[46] 21
2.8 Gated Recurrent Unit,variation of RNN, where x_t is input vector, h_t is output vector, and h_{t-1} is hidden state at previous timestep. Image from towardsdatascience 24
2.9 Bi-GRU structure diagram 25

3.1 A converged shape overlays a truth boundary for C2/C3 vertebrae with the ASM model 27
3.2 vision transformer proposed in Deep Learning and Cloud-Based Computation for Cervical Spine Fracture Detection System by Pawel Chald [18] 30

4.1 The frequency of fractures in the cervical spine and each vertebra 34
4.2 The metadata, extracted from a DICOM file related to one patient. 36

4.3 From left to right, the first one is the axial view, the middle one is the coronal view (anterior or posterior), and the last one is the sagittal view (side) of the anatomical space of a human. 37
4.4 Steps of image processing, mask acquisition, and fracture detection 38
4.5 The pseudo-code of U-Net segmentation model 40
4.6 AdamW (Adam with decoupled weight decay) algorithm by Loshchilov et. al. Source from [31] 43
4.7 A cosine annealing scheduler to adjust learning rate for semantic segmentation model 43
4.8 The required steps to detect fractures of the cervical spine 44
4.9 From left to right, the first one is the cropped image of one vertebra of the cervical spine related to study instance UID '1.2.826.0.1.3680043.1363' and the second image is its predicted mask by the proposed segmentation mask 45
4.10 CNN + Bi-GRU model architecture to detect fracture in the cervical spine 46
4.11 The defined model for the classification stage 47

5.1 the main architecture of the proposed 3D segmentation model 49
5.2 Learning curves of five folds 53
5.3 Examples of cervical spine CT scans and the predicted segmentation mask from the proposed semantic segmentation model on the validation dataset - sagittal plane . 55
5.4 predicted segmentation mask for four selected images from the test set in axial view 56

6.1 Three samples of the predicted bounding box around the vertebra by Yolov5 model - from left to right, the first one is the axial view of one vertebra of the cervical spine, the middle one is the predicted bounding box around that vertebra by Yolov5 and the last one is the predicted bounding box around that vertebra for the related segmentation mask. 62
6.2 one sample of the cropped cervical vertebra CT scan with corresponding mask . . 63
6.3 An overall view of the proposed classification model 66

6.4 detected fractures within a single vertebra by the proposed model 69
6.5 flase positive errors in the model related to patient IDs '1.2.826.0.1.3680043.24606', '1.2.826.0.1.3680043.2243','1.2.826.0.1.3680043.10005' 71

List of Tables

3.1 Comparison between the different methods in the detection of cervical spine fractures. 28

5.1 Achieved results showing the lower and upper bound of the learning rate, average training loss, average validation loss, and F1 score on cervical spine segmentation for each k-fold cross-validation of the studied data set 54

6.1 Augmentation techniques and their settings used in the proposed classification model 64

6.2 Predicting fracture probability for two study instances 68

6.3 Classification performance results on the dataset with 6299 negative cases and 701 positive cases extracted from the proposed model. TPR: sensitivity, TNR: Specificity, PPV: Positive predictive value, NPV: Negative Predictive Value, Acc: Accuracy, AUC: Area under curve. All the values are in percent 70

Acronyms

AI Artificial intelligence

ASM Active Shape Model

ASNR American Society of Neuroradiology

ASSR American Society of Spine Radiology

BCE Binary Cross Entropy

Bi-GRU Bidirectional GRU

CNN Convolutional Neural Network

CT computed tomography

DICOM Digital Imaging and Communications in Medicine

IoU Intersection-Over-Union

MRI Magnetic resonance imaging

NIFTI Neuroimaging Informatics Technology Initiative

Chapter 1

Introduction

This chapter introduces the problem studied, its significance, and the objectives of this research. In this chapter, a brief overview of the damage to the spinal area will be introduced. Then, the application of AI in the medical industry will be discussed. Then, the objective of this dissertation to detect the fracture in the cervical spine will be briefly presented in section 1.2.

1.1 problem statement and motivations

Motor vehicle accidents, sports-related injuries, and falling mostly result in damages to the spinal area, particularly, the upper cervical spine. For high-risk patients such as those with fractures or displacements seen on their plain radiographs, CT scans are taken to detect the location of fractures. It is vital to quickly diagnose and treat fractures to prevent neurologic deterioration and paralysis after trauma.

Artificial intelligence (AI) is being used in the healthcare industry in various areas such as cancer detection and analyzing MRI or X-ray images. Also, Convolutional Neural Network (CNN) [5], one type of deep learning algorithm, is particularly used in analyzing medical images. Image segmentation can be used to show the exact location of tumors or vertebrae in the human body. Previous studies in the segmentation of the cervical spine used models to detect the location of the spine and then find the boundary of each vertebra by deforming a 2D contour. Different machine learning approaches, such as Hough-based and U-Net architectures, are used with strong performances on semantic segmentation problems. U-Net is widely used in medical image segmentation, and it has been applied in many areas such as brain, liver, and cell segmentation. There are several different variations of the U-Net architecture that have been developed for specific tasks and applications.

This project develops a deterministic and automated model for the detection of cervical spine fractures, through the development, training, validation, and testing of a two-stage model. The first stage involves a semantic segmentation task, and the second stage involves a CNN model. The models use the patient's CT scans provided in the Digital Imaging and Communications in Medicine (DICOM) and Neuroimaging Informatics Technology Initiative (NIFTI) formats and then observe their performance metrics, such as dice coefficient, sensitivity, specificity, and accuracy [21].

It is expected the results of this research have a positive impact on the quick detection of cervical spine fractures before complications of undiagnosed injuries. Ultimately, it is expected that the deep learning models developed in this project will become useful models for research labs, clinical environments, and industries to correctly estimate the location of fractures in the cervical spine.

1.2 Objectives

The general objective of this research is to develop a system that can automatically and accurately identify disruptions or breaks in the bones of the cervical spine using medical imaging. In addition to the suitability of the proposed architecture as a standalone model, it can be used in an integrated framework to enhance robustness.

1.3 Outline of this Thesis

This dissertation is organized as follows. Chapter 2 presents background information about cervical spine anatomy and different types of injuries and disorders. Then, the recent studies to detect the disruptions in the cervical spine medical images and the application of AI will be discussed in Chapter 3. Chapter 4 highlights the analysis of the dataset and the requirements of training both segmentation and classification models. Both Chapter 5 and Chapter 6 present our proposed deep learning models. Finally, Chapter 7 summarizes the contributions of this research, provides the conclusions, and highlights future works to this thesis.

Chapter 2

Theory and Background Materials

This chapter first provides basic information on related topics to the main study of this dissertation in section 2.1. Then, diagnostic techniques of cervical spine fracture and image segmentation will be discussed that have been explored prior to the rise of deep learning for cervical spine fracture detection. Section 2.4 presents three significant deep-learning architectures that were the inspiration for this research: U-Net, CNN, and Bi-GRU.

2.1 Cervical spine anatomy

The cervical spine is the uppermost portion of the spine that runs from the base of the skull to the top of the thoracic spine. It is composed of seven vertebrae and is responsible for supporting the weight of the head, protecting the spinal cord, and allowing for a wide range of motion, including head and neck rotation, flexion, and extension. Understanding the anatomy of the cervical spine is important for medical professionals, researchers, and anyone who wants to maintain optimal neck health [12].

The cervical spine consists of seven vertebrae labeled C1 to C7. The first vertebra is an atlas with a ring-shaped bone that begins at the base of the skull. The primary movements of the atlas are flexion and extension, with a normal range of approximately 15° to 20° of flexion to hyperextension at the atlanto-occipital joint [62].

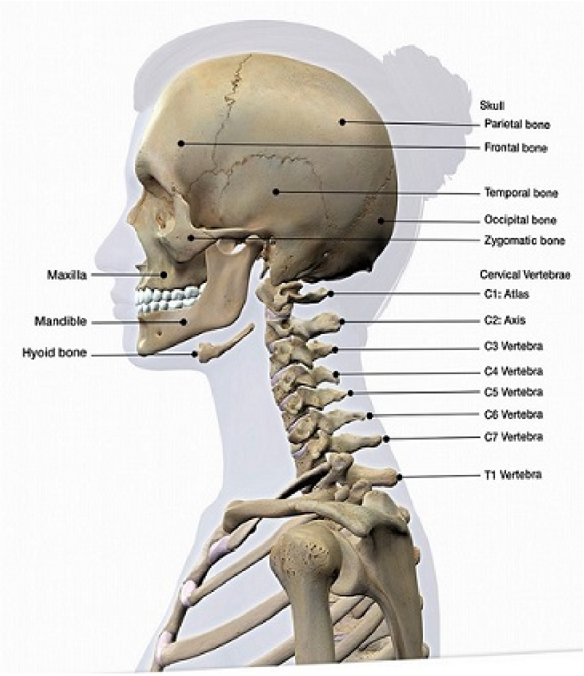


Figure 2.1: the anatomy of skull and cervical spine from greatbigcanvas [27]

The second vertebrae also called the axis, allows the atlas to pivot against it for the side-to-side 'no' rotation of the head. One of its prominent characteristics is the dens, a bony protuberance that extends superiorly from the body of C2. The dens act as a pivot point and form a critical articulation with the atlas, allowing for rotational movement of the head and neck. The typical range of rotation for the atlas on the axis is approximately 50° in each direction [62].

C3 to C7 vertebrae are structurally similar, but each possesses distinct features that contribute to their specific functions within the cervical spine. These vertebrae share common anatomical elements, including vertebral bodies, vertebral arches, spinous processes, transverse processes, and facet joints.

The cervical spine's overall flexion or extension is not necessarily reflected in the movement of individual vertebrae, and each vertebra contributes to motion differently. This distinct motion pattern is relevant for understanding injury mechanisms in the cervical spine [6]. The main structure of the cervical spine is shown in Figure 2.2.

Each cervical vertebra consists of a vertebral body, vertebral arch, and spinous process (Figure 2.3). The vertebral body is the anterior component of a cervical vertebra. It serves as the

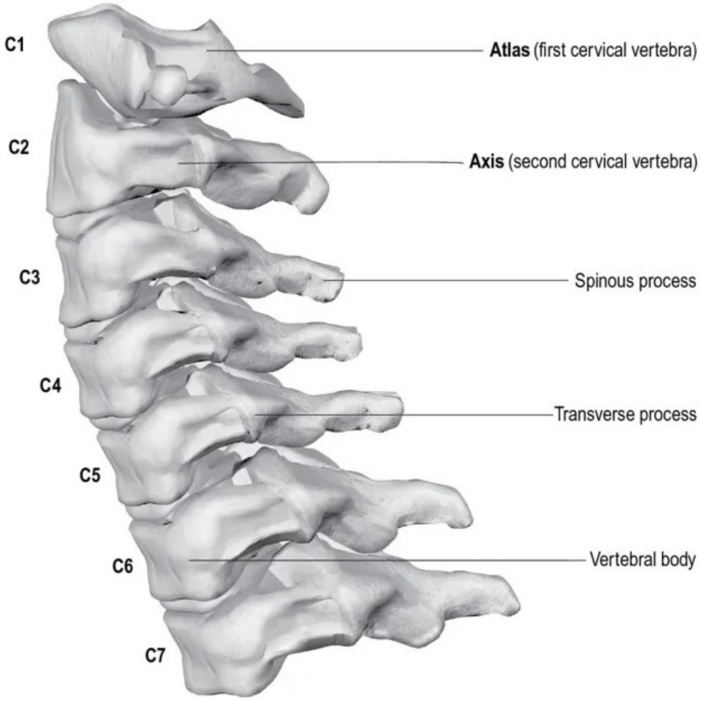

C1
Atlas (first cervical vertebra)
C2
Axis (second cervical vertebra)
C3
Spinous process
C4
Transverse process
C5
C6
Vertebral body
C7

Figure 2.2: the structure of the cervical spine, the uppermost portion of the spine from Musculoskeletal [61]

weight-bearing structure, responsible for providing stability and absorbing forces exerted on the spine. The vertebral bodies are connected through intervertebral discs, which act as cushions and allow for mobility. These cylindrical structures have a thick outer shell of compact bone, while the inner core consists of cancellous bone, giving them strength and resilience[19].

The cervical spine also includes intervertebral discs, which are located between adjacent vertebrae. They are essential components of the cervical spine, providing flexibility, shock absorption, and spinal alignment. During daily activities, the discs absorb and distribute the forces applied to the spine, reducing stress on the vertebrae and preventing damage to the spinal structures. Also, intervertebral discs allow for controlled movement between adjacent vertebrae, enabling the neck to bend, rotate, and flex in various directions. The discs contribute to maintaining the natural curvature and alignment of the cervical spine, ensuring optimal posture and balance [67].

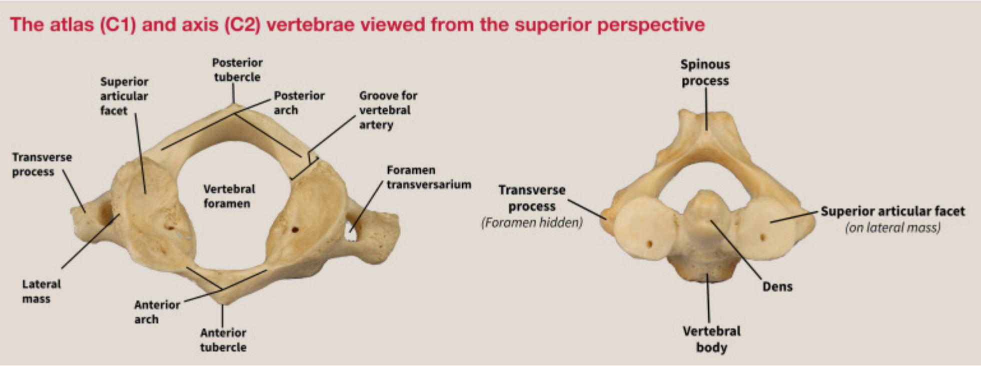


Figure 2.3: A superior view of one typical vertebra from Peter et al. [9]

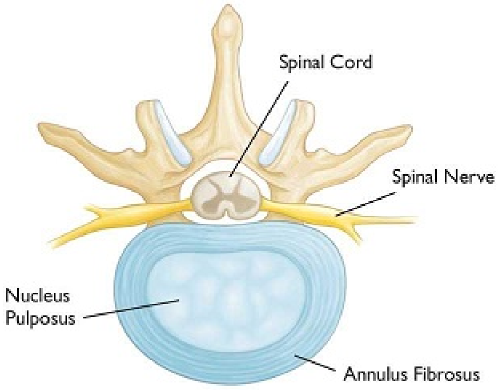


Figure 2.4: intervertebral view from the American Academy of Orthopaedic Surgeons [14]

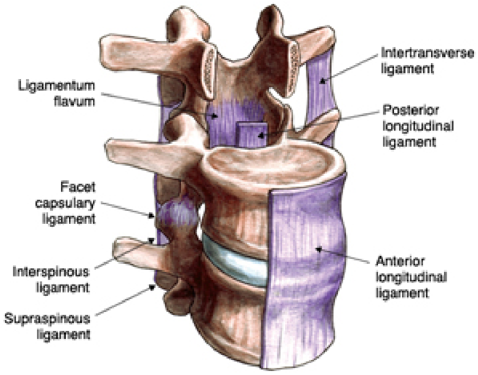


Figure 2.5: Spinal Ligaments from Health Central [65]

The proper functioning of the cervical spine is attributed to a network of ligaments that connect the vertebrae and provide structural integrity. Here, This research will be delved into the most important cervical spine ligaments, their functions, and their importance in maintaining spinal stability (Fig 2.5).

2.2 Cervical spine injuries and disorders

Injuries to the cervical spine can result from trauma, repetitive stress, or degenerative changes. These injuries can cause significant pain and discomfort and can lead to long-term disability. A cervical spinal cord injury is damage to the spinal cord in the neck region, which can result in a range of physical and neurological symptoms depending on the location and severity of the injury. The spinal cord is a long, thin bundle of nerve fibers that runs from the brain down through the spinal column and is responsible for transmitting signals between the brain and the rest of the body (Fig 2.6).

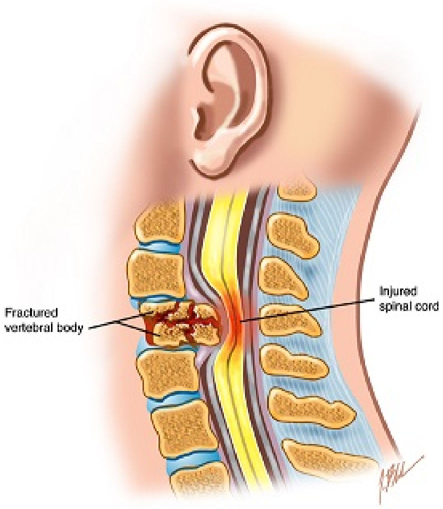


Figure 2.6: Spinal Cord injury from INSYNC PHISIO [52]

Whiplash is a common injury of the cervical spine that occurs when the neck is forced to move rapidly back and forth. This can happen in car accidents, sports injuries, or falls. Whiplash can cause damage to the muscles, ligaments, and nerves of the cervical spine, resulting in pain, stiffness, and limited range of motion. Symptoms of whiplash include neck pain, headache, dizziness, and fatigue. Treatment options for whiplash include rest, ice, heat, and over-the-counter pain medication. Physical therapy may also be recommended to strengthen the neck muscles and improve the range of motion. In severe cases, surgery may be necessary to repair damaged tissues [63].

Cervical sprains and strains are injuries to the muscles and ligaments of the cervical spine. These injuries can occur due to sudden trauma, repetitive stress, or poor posture. Symptoms of cervical sprains and strains include neck pain, stiffness, and limited range of motion [4].

Cervical radiculopathy is a condition in which the nerve roots that exit the cervical spine become compressed or irritated, causing pain, weakness, or numbness in the arms. This condition can be caused by a herniated disc, degenerative changes, or spinal stenosis. Treatment options for cervical radiculopathy include rest, over-the-counter pain medication, and physical therapy. In severe cases, surgery may be necessary to decompress the nerve roots and relieve pressure [42].

A cervical herniated disc is a common spinal condition that occurs when the outer layer of a spinal disc in the neck ruptures or tears, causing the inner gel-like material to leak out. This can lead to compression or irritation of the nerves in the neck, causing pain, weakness, and other symptoms. A cervical herniated disc can occur due to age-related wear and tear on the spinal discs or due to trauma or injury to the neck. Factors that increase the risk of developing a cervical herniated disc include aging, repetitive stress, poor posture, and genetics [53].

To conclude, injuries to the cervical spine can cause significant pain and discomfort and can lead to long-term disability. It's important to seek prompt medical attention if you experience any symptoms of cervical spine injury. Treatment options vary depending on the type and severity of the injury and may include rest, medication, physical therapy, or surgery.

2.3 Cervical Spine fracture detection

Cervical spine fractures are significant injuries that can have severe consequences if not diagnosed and managed promptly. They are commonly associated with traumatic events such as motor vehicle accidents, falls, and sports injuries. Timely detection and accurate diagnosis of cervical spine fractures are crucial to ensure appropriate treatment and prevent potential complications. In recent years, advances in medical imaging technology and innovative diagnostic techniques have greatly enhanced the detection of these fractures. This section explores the previous approaches and emerging technologies in cervical spine fracture detection.

2.3.1 Diagnostic Techniques:

Traditionally, the evaluation of suspected cervical spine fractures involved a combination of physical examination, plain radiographs, and computed tomography (CT) scans. Physical examination, including the assessment of range of motion and the presence of neurologic deficits, provides valuable initial information. However, it is not always reliable in identifying subtle fractures or injuries in unconscious or uncooperative patients. Radiography, CT, dual-energy CT (DECT), Magnetic resonance imaging (MRI), and Ultrasonography are some medical imaging procedures that are used in cervical spine fracture detection.

Radiography is the most commonly used method for detecting cervical spine fractures. The technique involves taking X-ray images of the cervical spine to identify any fractures or dislocations. Radiography is a non-invasive and relatively inexpensive method that can provide accurate results in most cases. However, it has some limitations, such as difficulty in visualizing soft tissue injuries and limited sensitivity in detecting subtle fractures [45].

computed tomography (CT) is another commonly used method for detecting cervical spine fractures. CT scans use X-rays to create cross-sectional images of the cervical spine, allowing for detailed visualization of bone and soft tissue structures. CT is more sensitive than radiography in detecting subtle fractures and can also identify soft tissue injuries. However, it exposes patients to higher levels of radiation than radiography [38].

dual-energy CT (DECT) is an advanced form of CT scanning that involves acquiring images at two different energy spectra simultaneously or near-simultaneously. DECT uses different peak energies for the low-energy and high-energy spectra, typically 80-100 kVp and 140-150 kVp, respectively. Dual-source DECT systems consist of two source-detector combinations, while single-source DECT with rapid kVp switching utilizes a single source-detector combination with fast switching between high and low tube voltages. These technologies enable spectral tissue characterization and provide improved soft-tissue contrast for various clinical applications [50].

Magnetic resonance imaging (MRI) is a non-invasive method that uses strong magnetic fields and radio waves to create detailed images of the cervical spine. MRI is particularly useful in detecting soft tissue injuries, such as ligamentous and disc injuries, which may not be visible on radiography or CT. However, it is less sensitive in detecting bony injuries and may not be suitable for patients with metal implants or claustrophobia [34].

Ultrasonography is a non-invasive imaging technique that uses high-frequency sound waves to create images of the cervical spine. Ultrasonography is particularly useful in detecting injuries to the spinal cord and nerve roots. However, it has limited sensitivity in detecting bony injuries and may not be suitable for patients with obesity or bowel gas [64].

In conclusion, each method has its advantages and limitations, and the choice of approach depends on the patient's clinical presentation and the suspected mechanism of injury. A combination of imaging and clinical evaluation is often necessary to make an accurate diagnosis and provide appropriate treatment.

2.3.2 Image segmentation

With the advancements in medical imaging technology, radiologists have access to high-resolution three-dimensional images of the cervical spine. However, manual interpretation of these images can be time-consuming and prone to human error. In recent years, image segmentation techniques have emerged as a valuable tool to aid in the accurate and efficient detection of cervical spine fractures.

Image segmentation is the process of partitioning an image into multiple regions, each of which corresponds to a different object or part of an object. There are various techniques for image segmentation, including thresholding, clustering, and contour detection[8].

Thresholding is one of the simplest and most commonly used methods for image segmentation. It involves selecting a threshold value and partitioning the image into two regions: one with pixel values below the threshold and the other with pixel values above the threshold. This method works well when the objects to be segmented have a distinct difference in intensity compared to the background[57]. Here are some commonly used thresholding methods:

Global thresholding involves selecting a single threshold value that applies uniformly across the entire image. This technique assumes that there is a distinct intensity value that can effectively separate the foreground and background. Common global thresholding algorithms include the popular Otsu's method [47], which automatically determines an optimal threshold by minimizing the intra-class variance.

Adaptive Thresholding In scenarios where lighting conditions or image properties vary across different regions, adaptive thresholding techniques are employed. These methods compute a local threshold value for each pixel based on its neighborhood. Adaptive thresholding accounts for variations in illumination, contrast, and noise levels, making it suitable for handling non-uniform backgrounds or unevenly illuminated images [54].

Clustering is a technique that involves grouping similar pixels in an image into clusters based on their color or intensity values. Cluster analysis can be used to segment images based on the similarity of their pixel values, which can be useful for segmenting objects with complex boundaries or for separating foreground and background pixels in an image[8].

Contour detection involves identifying the boundaries of objects in an image by detecting the edges or contours of the objects. Contour detection is often used in combination with other segmentation techniques, such as thresholding or clustering, to improve the accuracy and reliability of the segmentation[8].

Traditional image processing techniques such as thresholding, edge detection, and contour tracing are applied to the image to identify the edge of the spine and separate it from the surrounding tissue. However, these methods can be sensitive to variations in image quality and can be difficult to train[10].

2.4 Deep Learning and its architectures

This section, the main concepts of deep learning and its three noteworthy models that inspired the deep learning models developed in Chapters 3, 4, and 5 are discussed in greater detail.

2.4.1 Deep learning

Deep learning is a subset of machine learning that focuses on the development and application of artificial neural networks to model and understand complex patterns and relationships in data by using multiple layers of interconnected nodes, or artificial neurons, inspired by the structure of the human brain. These networks can be trained using supervised and unsupervised learning methods [15], depending on the available data and the specific application. Deep learning has shown remarkable success in a wide range of tasks, including image and speech recognition, natural language processing, and decision-making [16].

Training a deep neural network involves the following steps:

- **Data collection, data preprocessing, and feature engineering**

  The first step in training a deep neural network is to prepare the data. High-quality and representative data is crucial for the success of a deep learning model. By leveraging exploratory data analysis (EDA), we can obtain vital insights including identifying trends in time and data, removing data outliers, creating the correlation matrix, and uncovering patterns related to the target.

- **Assign Appropriate Model**

  The next step is to design the architecture of the neural network. Assigning an appropriate model depends on the nature of the problem being solved (classification, Regression, Image recognition, natural language processing problems and etc), the type of data being used (numerical or categorical data), and it involves deciding on the number of layers and neurons in each layer, the activation functions and etc.

- **Define Loss Function**

The loss function is used to measure the difference between the predicted output and the actual output of the network for a given input. The goal of the loss function is to provide a measure of how well the model is performing and to guide the optimization process during training. The choice of loss function depends on the nature of the problem being solved and the type of output being predicted. Some common loss functions include mean squared error, cross-entropy, and binary cross-entropy loss functions.

- **Define Optimizer**

  During the training process, the optimizer adjusts the weights and biases of the model based on the gradients of the loss function. The gradients provide information about the direction in which the parameters should be updated to reduce the loss. The optimizer uses this information to adjust the model's parameters in a way that reduces the loss function, and thus improves the performance of the model. There are various types of optimizers available, each with their own strengths and weaknesses. Some of the most commonly used optimizers include stochastic gradient descent (SGD), Adam, RMSprop, and Adagrad [16].

- **Model Training**

  During training, the neural network updates the weights and biases of the neurons in each layer to minimize the loss function. The process of model training can be an iterative process, where the model is fine-tuned until it reaches a satisfactory level of performance.

- **Validation**

  Once the model is trained, it is validated through a separate set of data that the network has not seen before. This helps to determine if the model is overfitting or underfitting the training data[23].

  Overfitting occurs when the model performs well on the training data but performs poorly on unseen data, which means it has not learned to generalize well. There are several techniques to address overfitting such as regularization, early stopping,s and data augmentation[40].

On the other hand, underfitting can occur when the model may not perform well on either the training data or the new data. To address underfitting, it is necessary to increase the model's complexity or the number of training data to help the model to learn the underlying patterns in the data more effectively.

- **Tuning the model**

  If the model is not performing well, it may be necessary to tune the model by adjusting the hyperparameters (learning rate, the number of hidden layers, activation function, batch size and etc) and other parameters used in the model architecture.

- **Deploying the model**

  Once the model has been trained and evaluated, it can be deployed for use in real-world applications.

Data augmentation is a technique used in machine learning and computer vision to increase the diversity and quantity of training data by applying various transformations or modifications to the existing dataset. The goal of data augmentation is to improve the performance, generalization, and robustness of machine learning models.

The process involves creating new artificial training examples by applying a set of predefined transformations or modifications to the original data samples. These transformations are designed to simulate realistic variations in real-world data and introduce different perspectives or views of the same underlying information. By augmenting the dataset with these modified examples, the model becomes exposed to a wider range of variations and can learn more robust and generalized patterns.

Data augmentation techniques can be applied to various types of data, including images, text, audio, and more. Common data augmentation techniques for image data include resizing, cropping, rotation, flipping, adding noise, adjusting brightness/contrast, and changing color channels. For text data, techniques like random word replacement, shuffling, and synonym replacement can be used. Audio data can be augmented by adding noise, changing pitch, or applying time stretching, among others.

The key advantages of data augmentation are:

- **Increased dataset size**

  By generating new examples, data augmentation effectively increases the size of the training dataset, allowing models to learn from a larger and more diverse set of samples.

- **Improved generalization**

  Data augmentation introduces variations and perturbations in the data, which helps models to generalize better to unseen or real-world scenarios. It reduces overfitting by exposing the model to different perspectives of the same data.

- **Robustness to variations**

  By simulating realistic variations in the data, data augmentation helps models become more robust to changes in lighting conditions, scale, rotation, and other factors that might be encountered during inference.

Data augmentation is widely used in deep learning and is particularly effective when the available training data is limited. It has become a standard practice to apply data augmentation alongside other techniques like regularization and dropout to enhance model performance and address common challenges in machine learning tasks.

2.4.2 Semantic segmentation and U-Net

Semantic segmentation is a fundamental task in computer vision that aims to assign meaningful labels to every pixel in an image. Unlike image classification, which predicts a single label for the entire image, semantic segmentation goes a step further by providing a detailed understanding of the visual content at a pixel level. It plays a crucial role in various applications, including autonomous driving, robotics, and medical imaging.

Semantic segmentation and Instance segmentation are applications of image segmentation. Instance segmentation goes further by classifying each pixel in addition to identifying each object in the image, whereas semantic segmentation does pixels-wise classification based on object category [59].

U-Net is a convolutional neural architecture designed for semantic segmentation in biomedical images and it was first introduced in a paper by Ronneberger et al. in 2015[46]. Three areas of particular concern in the field of medical imaging segmentation have been addressed by this network. First, there is a lack of big datasets in this area. The U-Net architecture has been very popular due to its high performance and ability to work well with small amounts of data. The traditional feed-forwarding CNN with fully connected layers needs a large number of parameters to learn, therefore requires considerable large datasets. Replacing the fully connected layer with up convolutional layer on the decoder side, U-Net architecture has much fewer learnable parameters than a fully connected layer. The second issue is to precisely capture local details, handle varying image sizes, and leverage skip connections for feature fusion. Last but not least, the noteworthy problem of objects of the same class overlap with each other has been alleviated by using a weighted loss to separate background labels between touching segments.

U-Net is characterized by its “U” shape, with a contracting path to capture context and a symmetric expanding path that enables precise localization. The contracting path is referred to as the encoder, consists of a series of convolutional, max-pooling, and rectified linear unit (ReLU) layers for feature extraction, while the expanding path referred to as the decoder, involves convolution,and concatenation layers, and ReLU activation function to up-sampling the

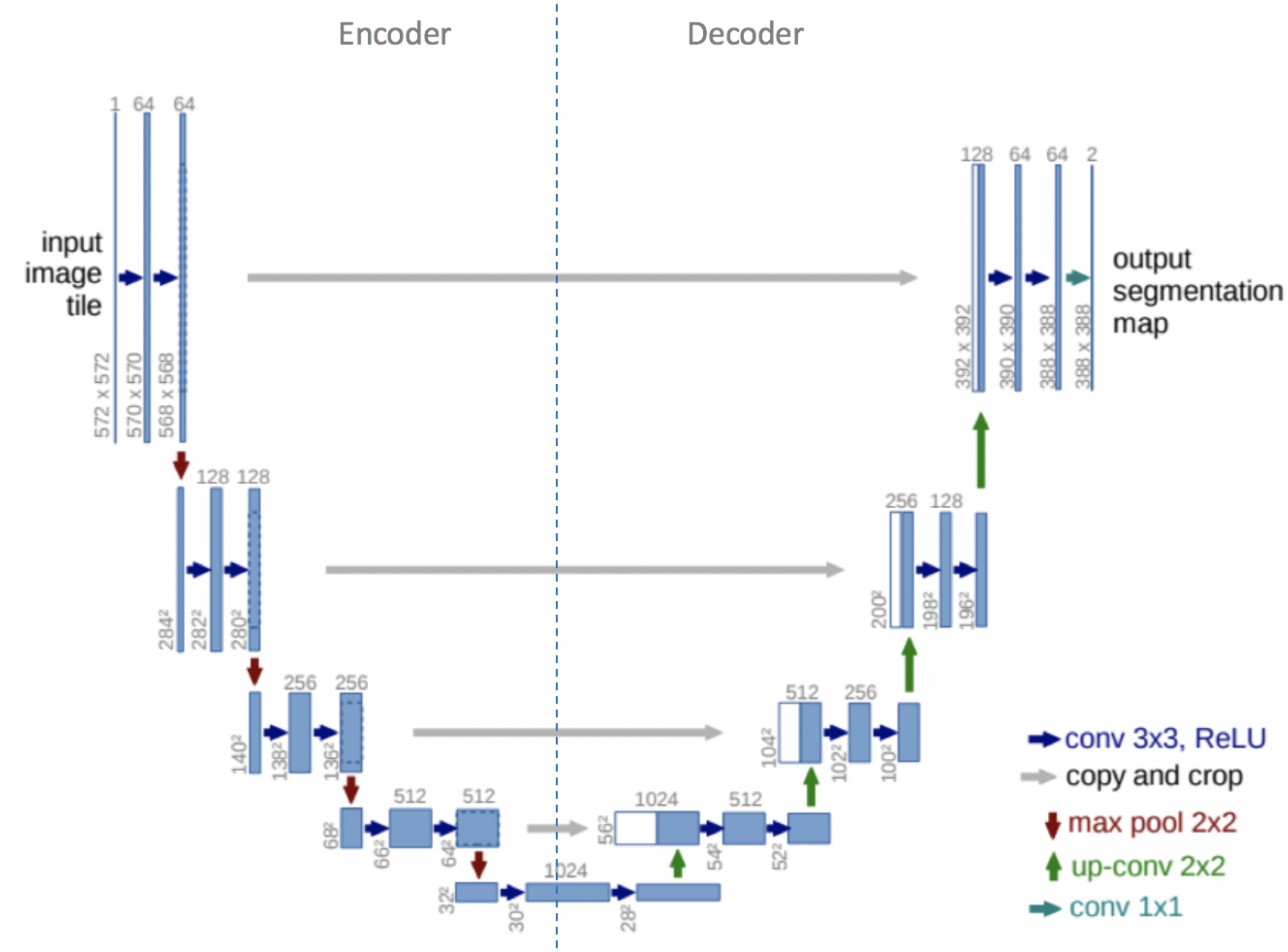


Figure 2.7: Architecture of U-Net based on the paper by Ronneberger et.al[46]

extracted feature maps. It also introduces skip connections that connect the contracting path with the corresponding expanding path at various resolutions. These skip connections allow the model to preserve and combine features from different scales. By directly concatenating feature maps from different levels, U-Net can exploit fine-grained details from early layers while incorporating high-level context from later layers. This aids in precise localization and improves segmentation accuracy, especially for small objects or fine structures. Figure 2.9 represents the structure of U-Net architecture.

The medical images and their corresponding segmentation maps are given to the U-Net to train the model. Ronneberger et al. [46] discussed several key aspects of training the U-Net model in their article. The authors emphasized the importance of weight initialization in controlling the learning process. They recommended initializing the weights for each feature map in the U-Net architecture to have unit variance. They suggested drawing the initial weights from a Gaussian distribution to achieve this. Also, they utilized a combination of a pixel-wise softmax function and a cross-entropy loss function for the final segmentation output. This combination helps optimize the model's performance in segmenting the input images accurately. As well as, due to the limited size of medical image datasets, the authors highlighted the significance of data augmentation techniques such as shifting, rotating, and adding noise to improve the generalization and robustness of the model.

As a result, this thesis leverages the semantic segmentation-based deep learning U-Net framework.

2.4.3 CNN with Bidirectional GRU layer

One possible approach for developing a fracture detection model in the cervical spine is to use a CNN. A CNN is a deep learning model to extract features from an input image and then uses those features to classify the image into different categories.

The CNN can be designed with multiple layers that perform different operations, such as convolution, pooling, and fully connected layers. The convolutional layers are responsible for detecting features in the input images with a single map for the grayscale image or a three map for a color image, while the pooling layers are used for classification tasks [16]. The convolutional layer applies a set of learnable filters to the input data, which are used to extract features from the input. The filters are usually 3×3 or 5×5 in size and applied in a sliding window fashion. The filters are convolved with the input data to create a set of feature maps to represent the presence of certain fractures in the input images. Fully connected layers then use the extracted features to make a final prediction to classify the input image.

The addition of a Bidirectional GRU layer allows the model to take into account not only the current input but also the previous and future inputs in the sequence.

GRU stands for "Gated Recurrent Unit", a type of recurrent neural network (RNN) architecture, which consists of repeated blocks of several convolutional layers with skip connections between them followed by a pooling layer that reduces the dimensions of the output. GRU was introduced by Kyunghyun Cho et al. in 2014 [37]. GRUs are similar to traditional RNNs but have a more complex structure that allows them to capture long-term dependencies in sequential data more effectively. The key feature of GRUs is their use of gating mechanisms to control the flow of information within the network. Specifically, GRUs have two gates: a reset and an update gate. The reset gate determines how much of the previous state should be forgotten, while the update gate determines how much of the current input should be used to update the current state. By dynamically updating and resetting the hidden state in this way, GRUs can selectively retain or discard information from previous time steps. The structure of GRU is shown in figure 2.10.

A Bidirectional GRU (Bi-GRU) [17], similar to the GRU model, consists of two GRU layers that

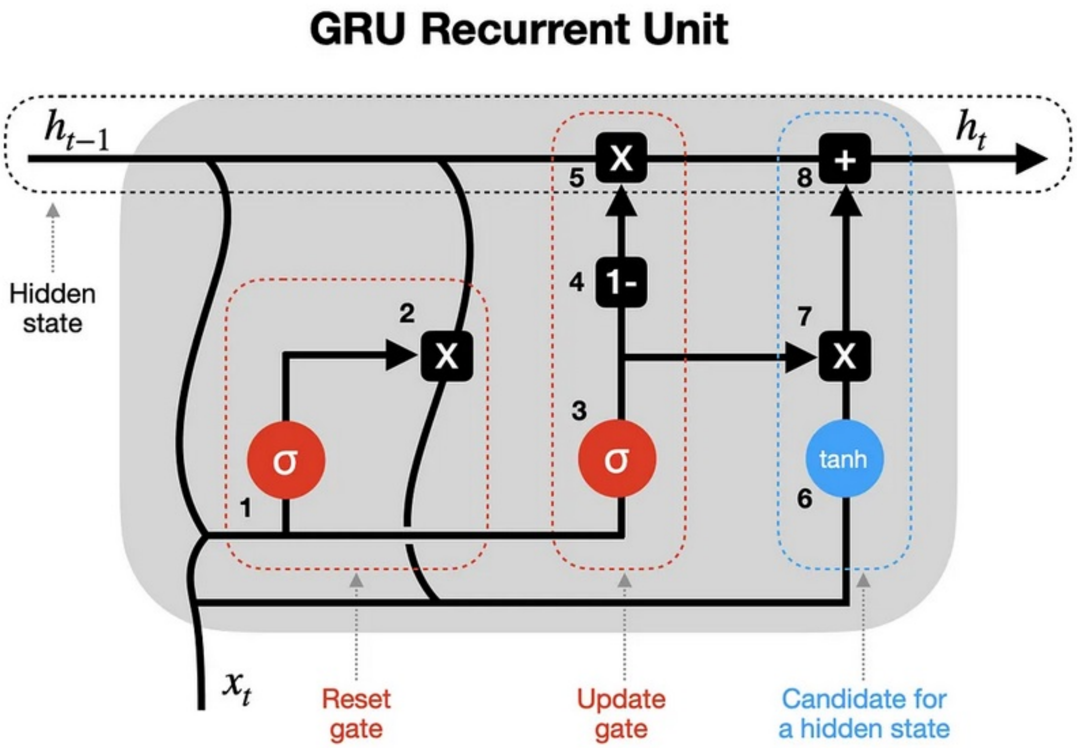


Figure 2.8: Gated Recurrent Unit,variation of RNN, where $\mathrm{x_t}$ is input vector, $\mathrm{h_t}$ is output vector, and $\mathrm{h_{t\text{-}1}}$ is hidden state at previous timestep. Image from towardsdatascience

process the input sequence in a positive time series and a reverse time series. The output of each GRU layer is concatenated to form the final output of the network. The forward or positive layer processes the input sequence from beginning to end, while the backward layer processes it from end to beginning. This allows the network to capture information from both past and future contexts at the same time. Figure 2.11 represents forward status and backward status in BI-GRU structure. In this research, a Bi-GRU is used.

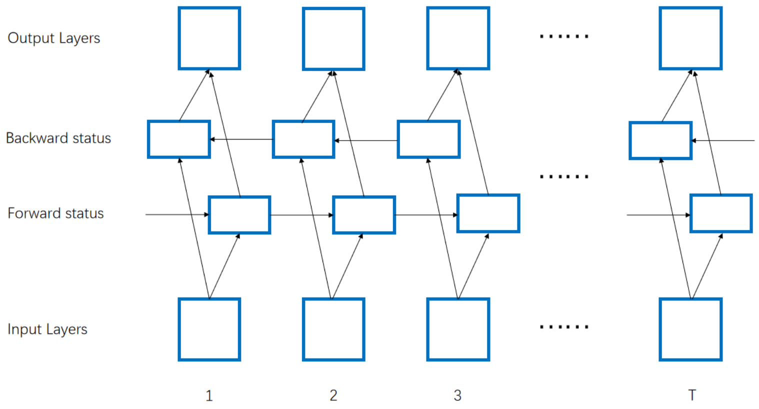


Figure 2.9: Bi-GRU structure diagram

The features extracted from the semantic segmentation model will be used as an embedding for a deep convolutional neural network (DCNN) with a bidirectional GRU layer to predict whether a given study has any fracture or not.

Chapter 3

Literature review

This chapter reviews different techniques that have been explored previously to detect cervical spine fractures in medical images.

3.1 Previous approaches

Early studies on segmenting the cervical spine in medical images used the Active Shape Model (ASM), a statistical method to model and track the shape and appearance of an object in an image, in order to search through digitized spine X-rays to identify specific vertebra within the images [39].

To assess the performance of ASM on the X-ray images, the research focused on one boundary area (C2/C3) and one complete vertebra (C3) and used a leave-one-out cross-validation technique. This involved selecting a single image from the dataset and using ASM to locate the C2/C3 boundary or the C3 vertebra in that image while testing the model on all the other images in the dataset. This was repeated for each image in the dataset, and the results were compared with manually determined ground truth values.

Nevertheless, ASM has only segmented the cervical spine C1 to C3, where the boundaries of the bones are clear, and the stacking effect of bones is not present. A mean point-to-point error of 0.2-0.3mm was achieved on 10000 X-rays images within this approach. Also, the performance of the ASM algorithm could be improved by refining the shape model and grayscale model and by incorporating additional information such as the orientation of the spine in the image. Figure 3.1 displays a good convergence for C1/C2 with the ASM method.

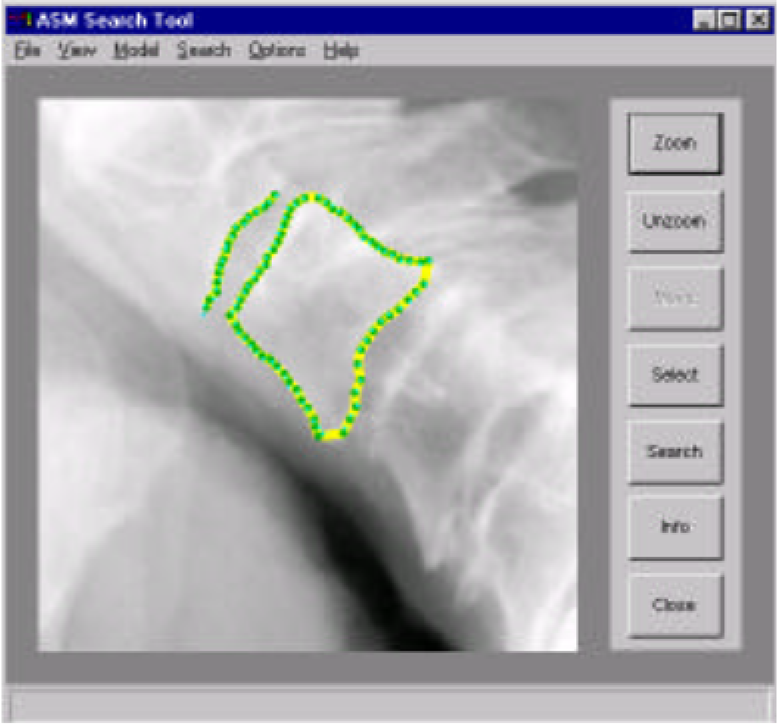


Figure 3.1: A converged shape overlays a truth boundary for C2/C3 vertebrae with the ASM model

Previous studies in the automatic localization of vertebrae were related to machine learning frameworks such as Hough transform-based and Random forest-based approaches on x-ray images and CT scans[26][55]. Within the random forest system, contextual information is incorporated with a supervised non-linear regression forest algorithm[1] to provide a fast estimation of vertebrae centers. Then, in the second stage, a joint model is used to provide a refined localization as well as vertebrae identification. It was shown that a localization error of less than 20mm with an overall success rate of 81% was achieved in the random forest method for vertebra identification in 200 CT scans[26]. The Hough forest framework included data-driven patch creation methods introduced to locate the vertebrae position, size, and orientation in 90 cervical x-ray images. This approach allowed the author to obtain an accuracy of 92.4% accuracy with an average mean error of 2.01mm for the spine detection task[55]. However, there are certain features such as the c-shape and stacked appearance of the cervical spines that make it difficult to distinguish between them. In recent studies[36][60], there has been a shift toward using more advanced techniques such as deep learning and CNNs to automatically identify ad segment the cervical spine in medical images.

U-Net, a deep learning architecture for fast and precise segmentation of images, can achieve high accuracy and robustness in segmenting cervical vertebrae in CT and MRI images[46]. A web-based automatic spine segmentation based on the U-Net architecture was developed by Kim et al [36]. A flask server framework was implemented for providing accessibility over the web in Python. This approach obtained a dice coefficient of 90.4% for the cervical spine segmentation task.

The below table compared the results of the mentioned research methods to successfully identify the location of the cervical spine in different medical images.

Reference	Method	Medical Images	Cost Function	Cost value	Applied metric	Metric value
Cloud-based architecture[18]	ViT	2019 CT	cross-entropy	close to 0.01	accuracy	98%
Retrospective analysis on an FDA-approved CNN by Aidoc[60]	CNN	12000 CT	K coefficient	0.70-0.82	accuracy	92%
Web-based automatic spine segmentation[36]	U-Net	344 CT	Dice Coefficient	90.4%	accuracy	91.64%
Automatic localization in arbitrary field-of-view[26]	Random Forest	200 CT	Localization error	Less than 6mm	accuracy	81%
Patch-based corner detection[55]	Hough Forest	90 X-rays	Average mean error	2.01mm	accuracy	92.4%
Use of Shape models to search digitalized spine x-rays [39]	ASM	10000 X-rays	Mean point-to-point error	0.2-0.3mm	-	-

Table 3.1: Comparison between the different methods in the detection of cervical spine fractures.

On the other hand, CNN can help extract relevant features from the segmented images to detect the fracture in each vertebra and greatly improve the classification effect[66].

Small et al. [60] proposed a retrospective analysis of the predictions of an FDA-approved CNN developed by Aidoc for the presence or absence of cervical spine fractures on CT. The framework consists of a region proposal stage with a 3D fully convolutional deep neural network and a false-positive reduction stage. They compared their model with the radiologist's results and found out that their CNN model's accuracy was 92% compared to 95% for radiologists. The missing fractures for both methods were related to fractured anterior osteophytes, transverse processes, and spinous processes. However, their results were based on a dataset with a high fracture prevalence, which will need to be implemented in a low fracture prevalence dataset in routine clinical practice.[60]

By leveraging cloud technologies in the medical industry, we will be able to use the development of scalable tools, improve existing models, and facilitate the creation of medical datasets[18]. Pawel Chlad et al. [18] proposed a cloud-based system for training and evaluating of vision transformer architecture for damage detection in vertebral CT scans. This article discusses the Vision Transformer (ViT), which is a recent architecture in the fields of computer vision [3].

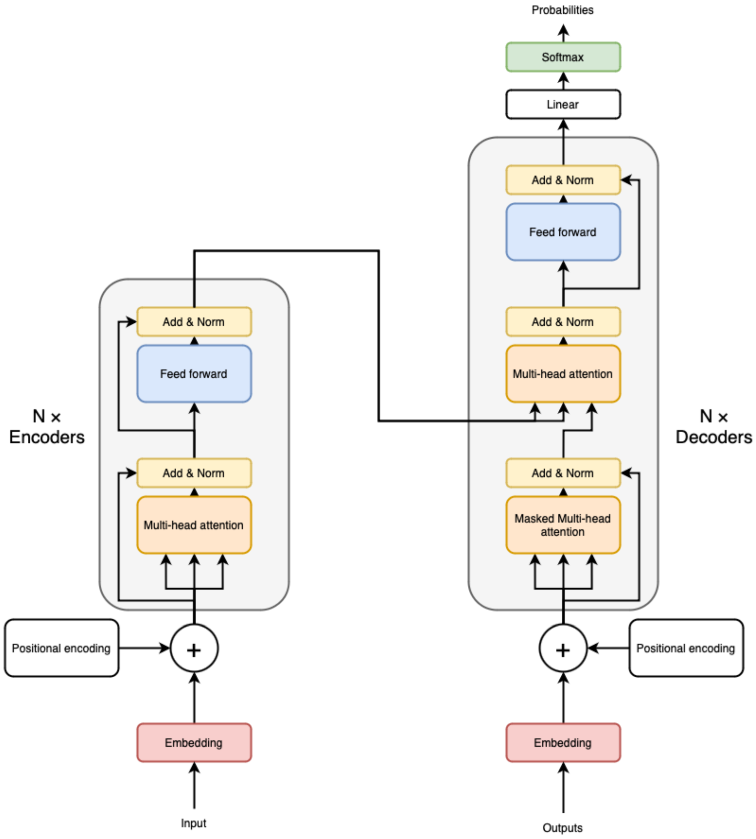


Figure 3.2: vision transformer proposed in Deep Learning and Cloud-Based Computation for Cervical Spine Fracture Detection System by Pawel Chald [18]

ViT is based on the transformer architecture and utilizes the mechanism of attention to focus on important parts of the input sequence. Unlike older architectures, attention in ViT is calculated per input rather than being statically encoded in the model itself.

The computational complexity of attention calculation is quadratic, which makes it computationally expensive. To address this, ViT uses an analysis space reduction technique by dividing the image into patches of equal size. These patches are flattened and turned into vectors, which are then encoded into a representation suitable for the transformer-encoder layer. Each encoding layer has the same fixed dimensionality, and the output of this projection is called patch embedding[18].

An encoder block in ViT consists of multi-head attention, add and norm, a feed-forward layer, and another add and norm layer. The multi-head attention calculates the importance of different parts of the input sequence using the scaled dot product attention function. Multiple attention layers operate in parallel, allowing the model to focus on different parts of the input sequence or image.

After passing through the multi-head attention layer, the input is added to the output, normalized, and passed to the feed-forward layer for projecting the input sequence into latent space. The parallel nature of the transformer and ViT architectures enables efficient evaluation on hardware accelerators. ViT reduces the inductive bias compared to convolutional neural networks (CNNs) and relies on learning spatial relationships during training (Figure 3.2).

In summary, the Vision Transformer architecture involves dividing an input image into patches, projecting them into latent space, adding positional encoding, passing through encoder blocks, and using an MLP head for producing the final prediction result. However, it emphasizes the importance of finding a balance between factors such as model size, bandwidth, latency, and data availability.

In conclusion, the above studies have sought to improve the accuracy of the provided models for medical image diagnosis to ensure that the patients are rapidly diagnosed before neurologic deterioration and paralysis after trauma. The present research explored U-Net and CNN for detecting vertebra fractures.

Chapter 4

Modelling and DataSet

This chapter provides all materials and methods used in this research. This chapter starts by introducing and analyzing the used dataset for this research. U-Net architecture, as a semantic segmentation model, will be discussed in section 4.2.1. A brief overview of the fracture detection model will be introduced in section 4.2.2.

4.1 Exploratory Data Analysis

4.1.1 Dataset analysis

Approximately 3000 CT scans of 2019 patients were collected by the American Society of Neuroradiology (ASNR) and American Society of Spine Radiology (ASSR) from 12 locations across six continents [22]. Spine radiology specialists from the ASNR and ASSR provided image annotations to indicate the presence, vertebral level, and location of any cervical spine fractures. The objective is to predict the probability of fracture for each of the seven cervical vertebrae denoted by C1, C2, C3, C4, C5, C6, and C7 as well as an overall probability of any fractures in the cervical spine.

Each CT scan consists of 100 to 800 slices of different thicknesses. Also, 47.60% of the given CT scans have fractures and 52.40% are normal images. Analyzing the data to know the prevalence of vertebrae fracture in each bone shows that the amounts vary from about 3.6% in C3 (lowest proportion of fractures) to about 19.5% in C7 (highest proportion). Figure 4.1 represents the overall percentage of the injured and intact cervical spine as well as the proportion of fractures in each vertebra of this study.

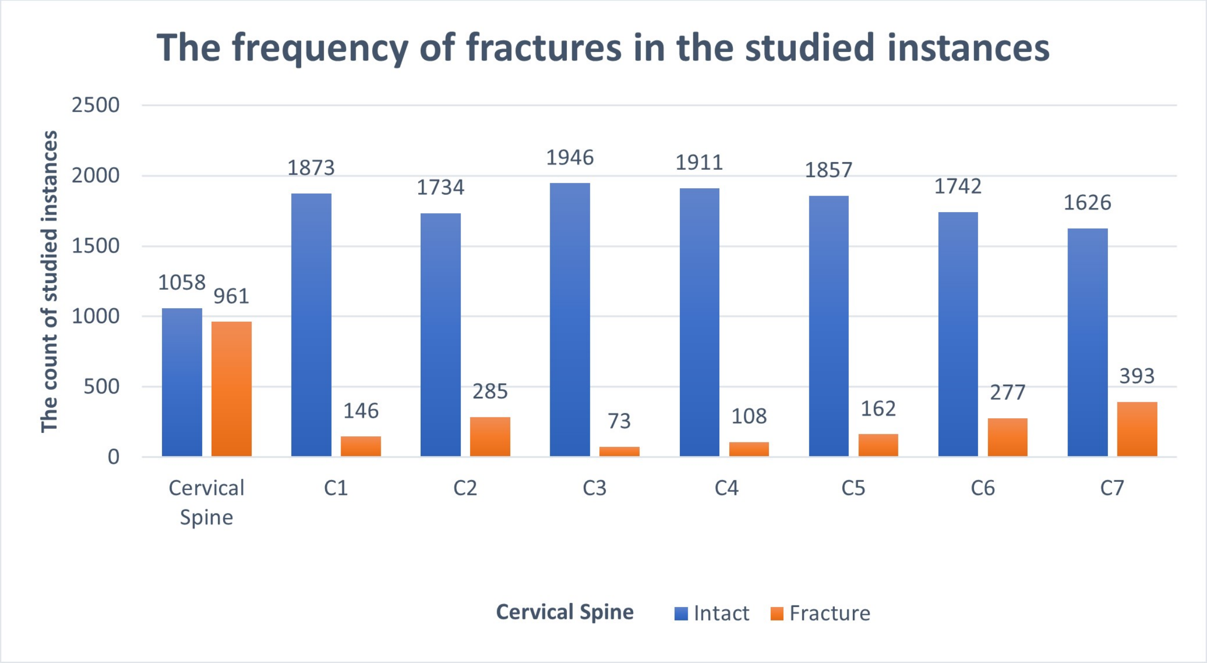


Figure 4.1: The frequency of fractures in the cervical spine and each vertebra

As mentioned, 961 out of the 2019 studied instances (47.60%) had fractures in their related CT scans. However, only 235 of the broken instances (24.45%) were annotated with bounding boxes and classified according to the cervical spine classification. Therefore, it is essential to apply a deep learning architecture that is suitable for medical imaging, where data availability can be limited.

On the other hand to improve the accuracy and robustness of the model, transforming the input images in various ways such as spatial transformations[29], and spatial-temporal transformations[28] can be used. Spatial transformations refer to a set of techniques that modify the input data by applying geometric transformations such as rotations, translations, scaling, and flips to the images in order to augment the data and improve the performance of a machine learning model. This has been done in this research by randomly applying the flipping spatial transformations and grid distortion to the images and adding these transformed images to the training set.

4.1.2 Image Format

Throughout this research, DICOM and NIFTI, the most popular medical image formats were used for storing and segmenting the images. Medical Images are visual representations of the body or specific parts of the body that are used for diagnosis and therapeutic purposes. The key properties of medical images are:

- Dimensionality

  The number of features or variables that are used to represent an image.

- Pixel Depth

  The number of bits required to encode each pixel is known as pixel depth.

- Photometric Interpretation

  It represents how the pixel data should be interpreted.

- Metadata

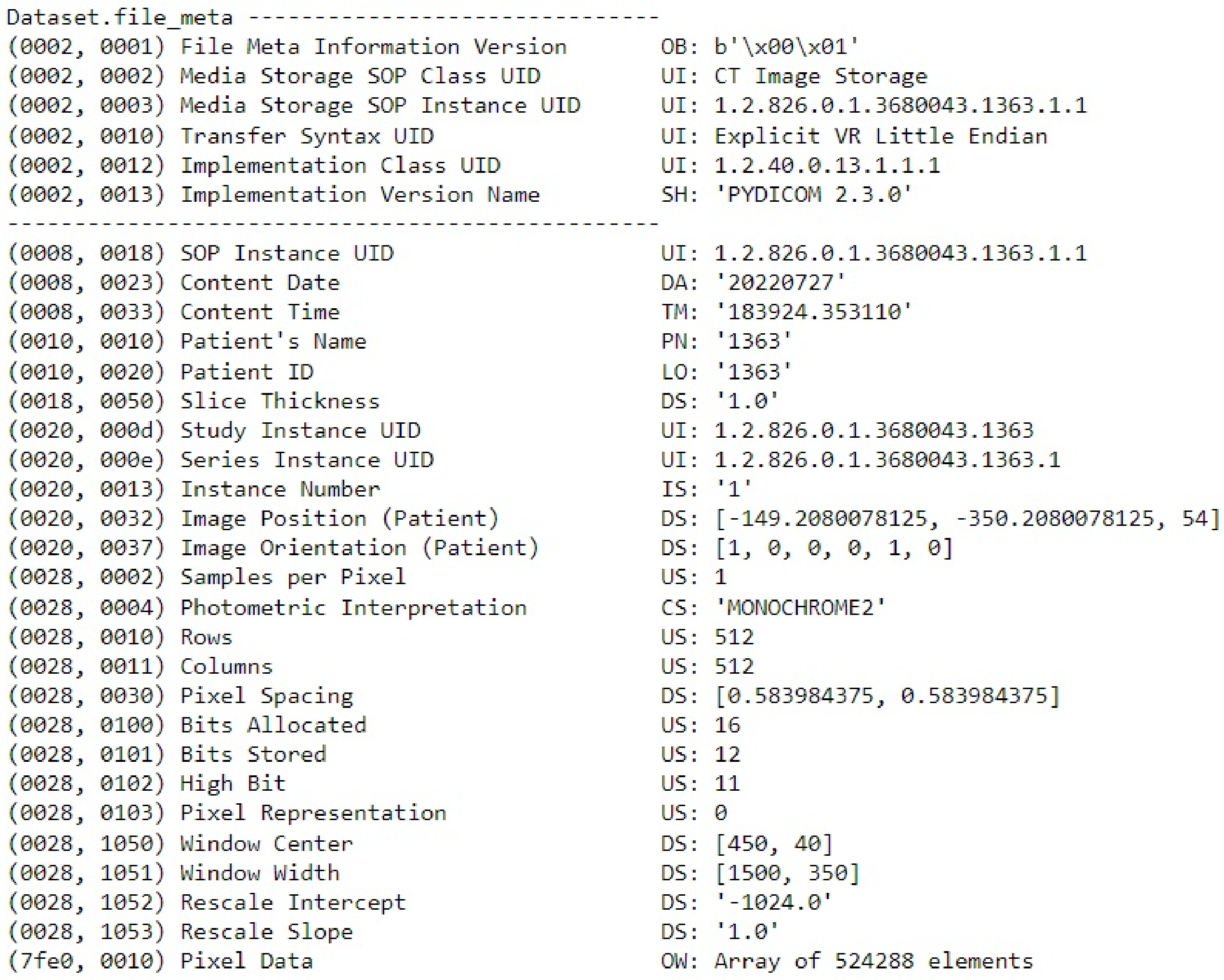
  It is used to annotate and exploit image-related information for research purposes.

- Pixel Data

  It is an array of values to store the pixels of an image[56].

The CT scans in this research store up to 2 bytes per pixel in their grayscale DICOM files. They are 3-dimensional volumes, where the third dimension is spatial to show different locations of scans.

DICOM is a standard for handling, storing, printing, and transmitting information in medical imaging. It was initiated by the National Electrical Manufacturers Association (NEMA) in the 1980s to address the need for a standard in medical images, such as X-rays, CT scans, and MRIs to handle and share them between different devices and systems[11]. DICOM organizes information into a dataset that contains metadata to provide information about the patient's name and identification, the date the image was taken, and the image data itself. Images in this research have resolutions of less than 1 mm deep and have an average of 512 pixels as width and 512 as height. An example of reading the metadata from a sample DICOM file is shown in Figure 4.2.



```
Dataset.file_meta -------------------------------
(0002, 0001) File Meta Information Version       OB: b'\x00\x01'
(0002, 0002) Media Storage SOP Class UID         UI: CT Image Storage
(0002, 0003) Media Storage SOP Instance UID      UI: 1.2.826.0.1.3680043.1363.1.1
(0002, 0010) Transfer Syntax UID                 UI: Explicit VR Little Endian
(0002, 0012) Implementation Class UID            UI: 1.2.40.0.13.1.1.1
(0002, 0013) Implementation Version Name         SH: 'PYDICOM 2.3.0'
-------------------------------------------------
(0008, 0018) SOP Instance UID                    UI: 1.2.826.0.1.3680043.1363.1.1
(0008, 0023) Content Date                        DA: '20220727'
(0008, 0033) Content Time                        TM: '183924.353110'
(0010, 0010) Patient's Name                      PN: '1363'
(0010, 0020) Patient ID                          LO: '1363'
(0018, 0050) Slice Thickness                     DS: '1.0'
(0020, 000d) Study Instance UID                  UI: 1.2.826.0.1.3680043.1363
(0020, 000e) Series Instance UID                 UI: 1.2.826.0.1.3680043.1363.1
(0020, 0013) Instance Number                     IS: '1'
(0020, 0032) Image Position (Patient)            DS: [-149.2080078125, -350.2080078125, 54]
(0020, 0037) Image Orientation (Patient)         DS: [1, 0, 0, 0, 1, 0]
(0028, 0002) Samples per Pixel                   US: 1
(0028, 0004) Photometric Interpretation          CS: 'MONOCHROME2'
(0028, 0010) Rows                                US: 512
(0028, 0011) Columns                             US: 512
(0028, 0030) Pixel Spacing                       DS: [0.583984375, 0.583984375]
(0028, 0100) Bits Allocated                      US: 16
(0028, 0101) Bits Stored                         US: 12
(0028, 0102) High Bit                            US: 11
(0028, 0103) Pixel Representation                US: 0
(0028, 1050) Window Center                       DS: [450, 40]
(0028, 1051) Window Width                        DS: [1500, 350]
(0028, 1052) Rescale Intercept                   DS: '-1024.0'
(0028, 1053) Rescale Slope                       DS: '1.0'
(7fe0, 0010) Pixel Data                          OW: Array of 524288 elements
```

Figure 4.2: The metadata, extracted from a DICOM file related to one patient.

On the other hand, spine radiology specialists performed manual segmentations to provide vertebra masks in the NIFTI format. NIFTI stands for "Neuroimaging Informatics Technology Initiative" and was developed by the National Institute of Health (NIH). Figure 4.3 represents three planes in the anatomical space of a human used to describe the different anatomical structures and movements. The sagittal plane divides the body into left and right halves, while the coronal plane divides the body into front and back halves and the axial plane divides the body into top and bottom halves. These three planes are important for understanding the different anatomical structures and movements of the human body. They are used in medical and scientific fields to describe and analyze various conditions, injuries, and diseases.
Figure 4.3 was extracted from the provided DICOM slices of one patient in this research. The key issue is that NIFTI files (which combine all DICOM files related to each study) consist of segmentation in the sagittal plane, whereas the DICOM files are in the axial plane. Therefore, a 3D semantic segmentation model was created for this research.

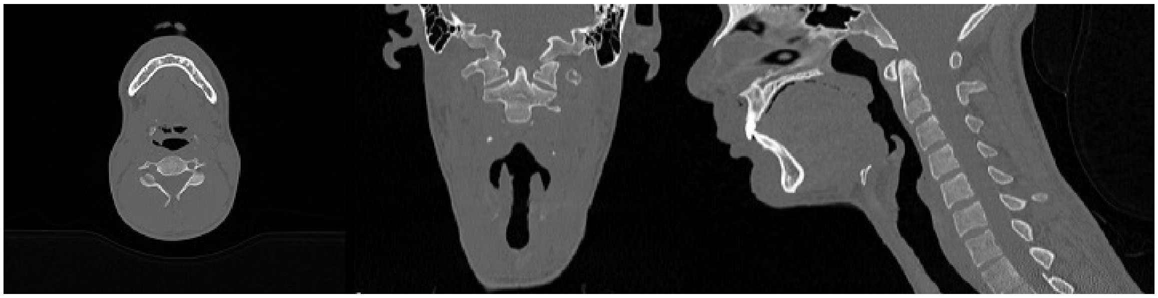


Figure 4.3: From left to right, the first one is the axial view, the middle one is the coronal view (anterior or posterior), and the last one is the sagittal view (side) of the anatomical space of a human.

Figure (4.4), extracted from the provided dataset, shows the process of vertebra segmentation and fracture detection. The given CT scans were in the size of 512 × 512 pixels. Therefore, a process of resizing images to the lower size of 128 × 128 pixels was implemented in the dataloader. The process of segmentation consisted of selecting the area that the radiologist identifies as the cervical vertebra (colored yellow in figure 4.4) and the remaining portion of the image considered as the background (colored purple in figure 4.4). Finally, a red bounding box is used for the fracture area.

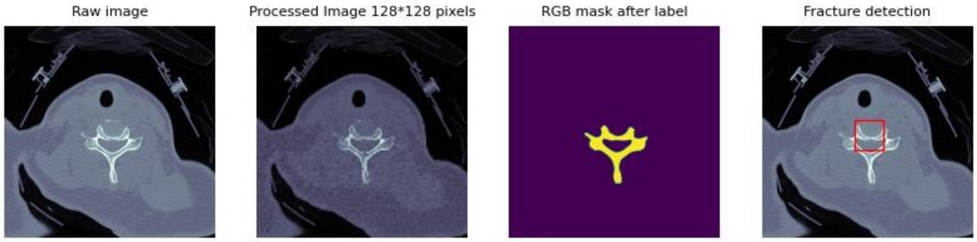


Figure 4.4: Steps of image processing, mask acquisition, and fracture detection

4.2 Model training

4.2.1 3D Semantic Segmentation model

Depending on the medical equipments, medical images can be 2D or 3D. The 2D images need less memory to store and can train faster on a segmentation model due to less number of learned parameters. On the other hand, 3D medical images provide spatial coherence with important information to improve the segmentation model performance [58]. This chapter will provide important considerations to implement an efficient 3D medical image segmentation model.

The semantic segmentation model was implemented using Python programming language and the machine learning libraries such as Torch, Monai, and Timm.

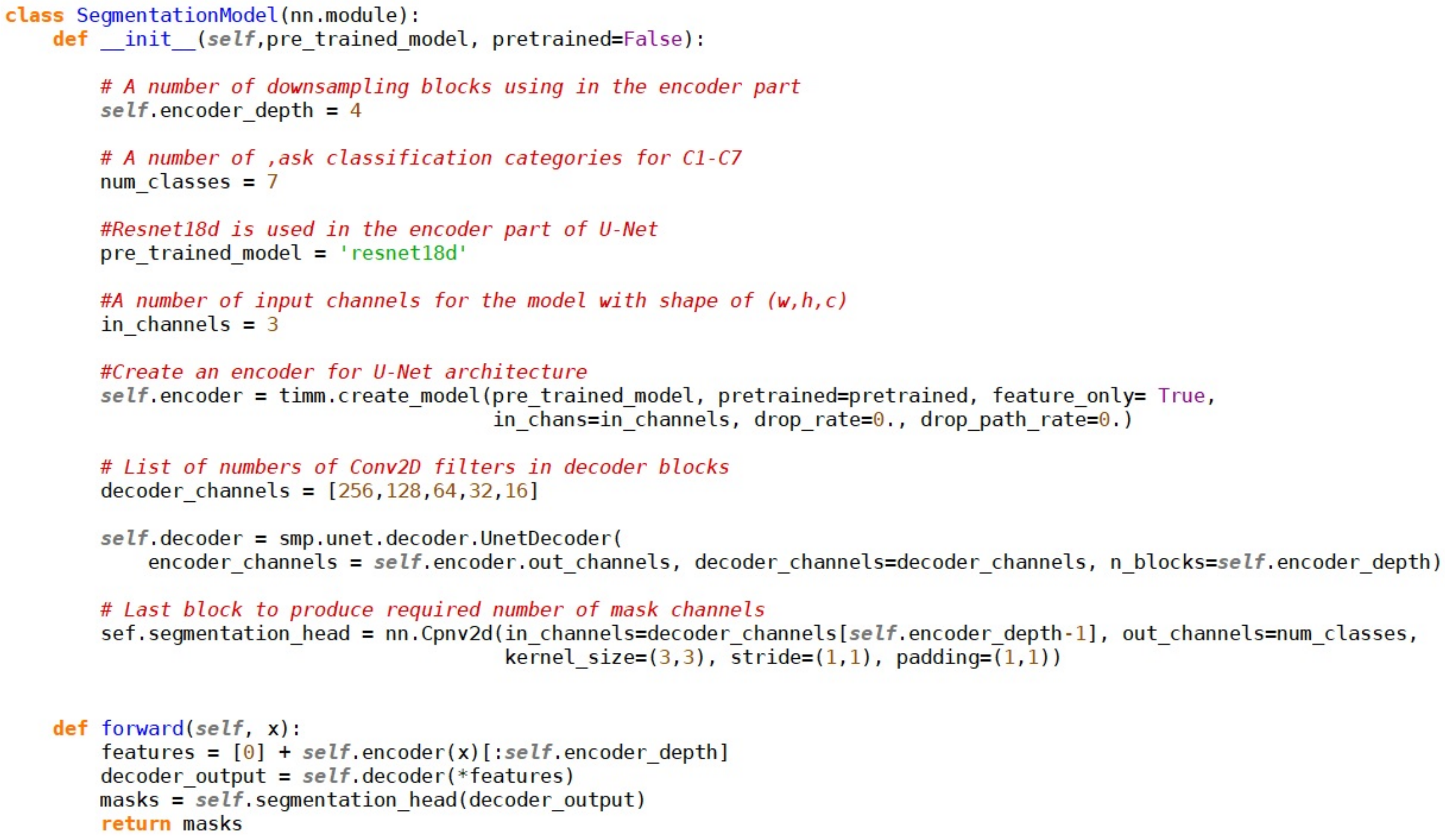
The CT scans and their respective segmentation masks were used as the input data for the model. The architecture was designed to accept input tensors of shape $128 \times 128 \times 128$, where the third dimension indicates the number of image slices. K_Fold Cross_validation was used to split the dataset into 5 subsets (k=5), and consider 4 subsets for training the model and one subset for the validation set to test the model's performance.

The U-Net architecture [46], a commonly used CNN for the purpose of biomedical image semantic segmentation, was implemented and this architecture was chosen based on its good performance on the small amount of data (only 87 of 2019 study instances were segmented by radiologists), which was reported in its paper[46]. U-Net consists of two pathways, an encoder and a decoder, for classifying and localizing each object in an image.

The encoder part involves multiple convolution, activation functions, and max-pooling layers to downsample data for feature extraction. A pre-trained ResNet18d, an 18-layer Residual network, was used for the encoder section of the U-Net to ease the training of the network and returned all extracted feature maps from each layer[35]. The ResNet consists of multiple convolutional layers, ReLU, and 3×3 max-pooling layers followed by 4 layers (blocks) with similar behavior. Each of the layers follows the same pattern and performs 3x3 convolution with a fixed feature map dimension[64, 128, 256, 512] respectively.

The decoder part involves upsampling the extracted feature maps with convolution, and concatenation for localization information. This research used the segmentation library in PyTorch to automatically construct the decoder section of U-Net based on the number of encoder channels.

The implemented model, through the encoder and decoder parts, extracted efficiently both the segmentation and location of cervical vertebrae for each studied CT scan. The following code shows how the U-Net segmentation model is defined in this research.



```
class SegmentationModel(nn.module):
    def __init__(self,pre_trained_model, pretrained=False):

        # A number of downsampling blocks using in the encoder part
        self.encoder_depth = 4

        # A number of ,ask classification categories for C1-C7
        num_classes = 7

        #Resnet18d is used in the encoder part of U-Net
        pre_trained_model = 'resnet18d'

        #A number of input channels for the model with shape of (w,h,c)
        in_channels = 3

        #Create an encoder for U-Net architecture
        self.encoder = timm.create_model(pre_trained_model, pretrained=pretrained, feature_only= True,
                                         in_chans=in_channels, drop_rate=0., drop_path_rate=0.)

        # List of numbers of Conv2D filters in decoder blocks
        decoder_channels = [256,128,64,32,16]

        self.decoder = smp.unet.decoder.UnetDecoder(
            encoder_channels = self.encoder.out_channels, decoder_channels=decoder_channels, n_blocks=self.encoder_depth)

        # Last block to produce required number of mask channels
        sef.segmentation_head = nn.Cpnv2d(in_channels=decoder_channels[self.encoder_depth-1], out_channels=num_classes,
                                          kernel_size=(3,3), stride=(1,1), padding=(1,1))


    def forward(self, x):
        features = [0] + self.encoder(x)[:self.encoder_depth]
        decoder_output = self.decoder(*features)
        masks = self.segmentation_head(decoder_output)
        return masks
```

Figure 4.5: The pseudo-code of U-Net segmentation model

The model was trained using a weighted sum of Dice loss and a binary cross-entropy to leverage the flexibility of Dice loss and the curve smoothing property of cross-entropy[33]. These both loss functions are defined as:

- Binary Cross Entropy (BCE)

  It is a commonly used loss function in binary classification problems, where the goal is to predict one of two outcomes (fractured or intact vertebra)[41]. It measures the dissimilarity between the true label and predicted probability distribution over two classes, and can be defined as:

  $L_{\mathrm{BCE}}(y, \hat{y}) = -(y \times log(\hat{y}) + (1 - y) \times log(1 - \hat{y}))$

  where $\hat{y}$ is the prediction value by model.

- Dice Loss

  It is a loss function commonly used in computer vision tasks and measures the overlap between the predicted mask and the ground truth[33]. It can be defined as:

  $\text{Dice Loss} = 1 - \frac{2 \times area-of-overlap-between-predicted-segmentation-and-ground-truth}{total-number-of-pixel-in-both-predicted-segmentation-and-ground-truth}$

The training process consists of training the segmentation model on 87 study instances with different numbers of CT scans or DICOM files. A process of resizing images to 128*128 pixels was implemented in the data loader. During the training process, the Dice loss and BCE in training and validation steps were recorded after each epoch within 5 folds. After the training process, the prediction process used the model for each data on the validation dataset to calculate the model performance since the validation dataset had ground truth segmentation masks.

Segmentation model performance was evaluated against Dice coefficient or F1 score as described by Shruti Jadon in 2020[33]. It is defined as:

$\text{Dice Coefficient} = \frac{2 \times area-of-overlap-between-predicted-segmentation-and-ground-truth}{total-number-of-pixel-in-both-predicted-segmentation-and-ground-truth}$

The dice coefficient is similar to the Intersection-Over-Union (IoU), which is one of the most commonly used metrics in semantic segmentation tasks[51]. It measures the overlap between the predicted segmentation and the ground truth mask. It is defined as:

$$IoU = \frac{TP}{FP + FN + TN}$$

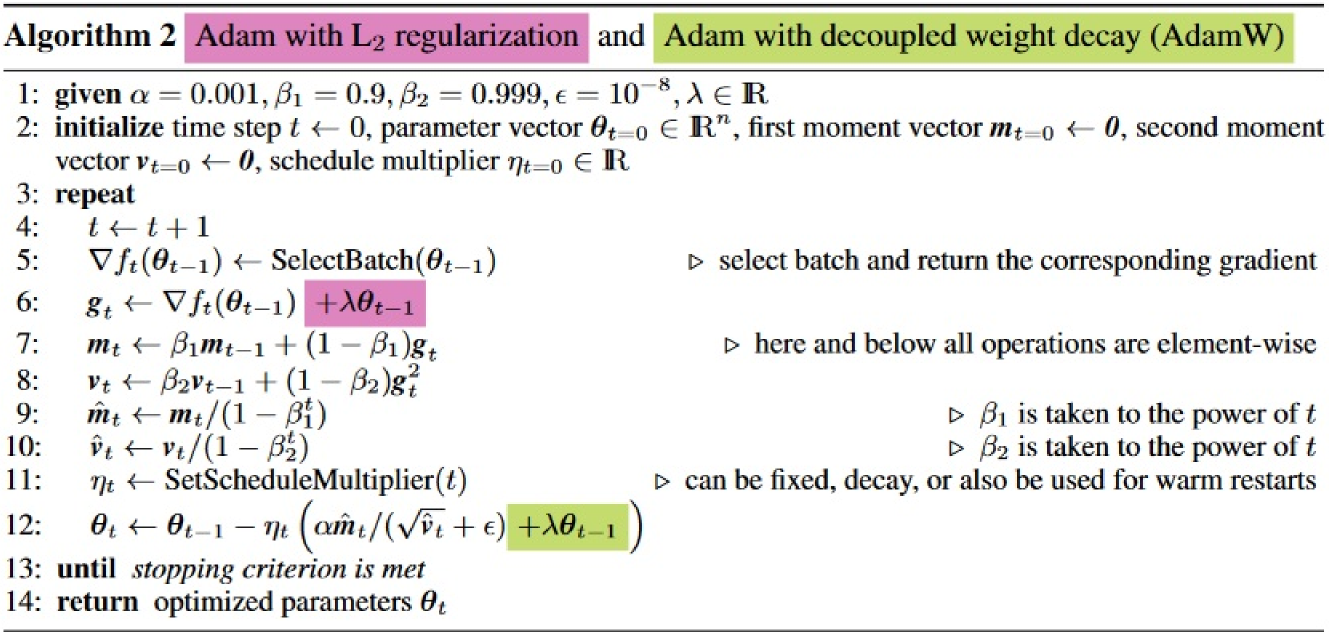
where TP, FP, and FN indicate the true positive, false positive, and false negative counts, respectively. The value of Dice coefficient and IoU ranges from 0 to 1, where a value of 1 indicates a perfect overlap between the predicted and ground truth masks, and a value of 0 indicates no overlap. A higher IoU value indicates a better performance of the segmentation algorithm.

In order to improve the model's accuracy and reduce the loss, AdamW was used as an optimizer in this research [31]. The Adam optimizer was presented by Diederik Kingma and Jimmy Ba [20] after that, the following AdamW algorithm (Fig 3.6) was used to provide better generalization performance than Adam with L2 regularization[31].

This algorithm receives α (learning rate),β_1 and β_2 (coefficients used for computing running averages of gradient and its square),θ_0(iterable of the parameter to optimize),f(θ)(maximize the parameter based on the objective),ϵ (term added to the denominator to improve numerical stability), and λ (weight decay coefficient) to render the optimal setting of the learning rate and improve regularization.

The following algorithm (Figure 4.6) shows the differences between Adam (purple background color around the text) and AdamW (green background color around the text). The main difference between AdamW and Adam optimization algorithms is that AdamW incorporates weight decay regularization ($\lambda\theta$) into the learning rate calculation, whereas Adam does not. Weight decay regularization is a technique used to prevent overfitting in neural networks by adding a penalty term to the cost function that encourages the neural network's weights to remain small.

Therefore, the AdamW optimizer was used with an initial learning rate of 0.0005 and a decay rate of 0.01 to minimize the loss function. Also, a cosine annealing scheduler with the following formula [32], was used to adjust the learning rate based on the number of epochs and improve



Algorithm 2 Adam with L_2 regularization and Adam with decoupled weight decay (AdamW)

```
given $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}, \lambda \in \mathbb{R}$
initialize time step $t \leftarrow 0$, parameter vector $\boldsymbol{\theta}_{t=0} \in \mathbb{R}^n$, first moment vector $\boldsymbol{m}_{t=0} \leftarrow \boldsymbol{0}$, second moment vector $\boldsymbol{v}_{t=0} \leftarrow \boldsymbol{0}$, schedule multiplier $\eta_{t=0} \in \mathbb{R}$
repeat
    $t \leftarrow t + 1$
    $\nabla f_t(\boldsymbol{\theta}_{t-1}) \leftarrow \text{SelectBatch}(\boldsymbol{\theta}_{t-1})$    ▷ select batch and return the corresponding gradient
    $\boldsymbol{g}_t \leftarrow \nabla f_t(\boldsymbol{\theta}_{t-1})$ $+\lambda\boldsymbol{\theta}_{t-1}$
    $\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + (1 - \beta_1)\boldsymbol{g}_t$    ▷ here and below all operations are element-wise
    $\boldsymbol{v}_t \leftarrow \beta_2 \boldsymbol{v}_{t-1} + (1 - \beta_2)\boldsymbol{g}_t^2$
    $\hat{\boldsymbol{m}}_t \leftarrow \boldsymbol{m}_t/(1 - \beta_1^t)$    ▷ $\beta_1$ is taken to the power of $t$
    $\hat{\boldsymbol{v}}_t \leftarrow \boldsymbol{v}_t/(1 - \beta_2^t)$    ▷ $\beta_2$ is taken to the power of $t$
    $\eta_t \leftarrow \text{SetScheduleMultiplier}(t)$    ▷ can be fixed, decay, or also be used for warm restarts
    $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \eta_t \left( \alpha \hat{\boldsymbol{m}}_t / (\sqrt{\hat{\boldsymbol{v}}_t} + \epsilon) +\lambda\boldsymbol{\theta}_{t-1} \right)$
until stopping criterion is met
return optimized parameters $\boldsymbol{\theta}_t$
```

Figure 4.6: AdamW (Adam with decoupled weight decay) algorithm by Loshchilov et. al. Source from [31]

the optimization process during the training (Fig 4.7).

$\eta_{\mathrm{t}} = {\eta^i}_{\mathrm{min}} + \frac{1}{2}({\eta^i}_{\mathrm{max}} - {\eta^i}_{\mathrm{min}})(1 + \cos(\frac{T_{\mathrm{cur}}}{T_{\mathrm{i}}}\pi))$

Where ${\eta^i}_{\mathrm{min}}$ and ${\eta^i}_{\mathrm{max}}$ are learning rate ranges, and $\mathrm{T}_{\mathrm{cur}}$ shows the number of performed epochs.

The decrease in the learning rate is shown in Figure 4.7 for this research.

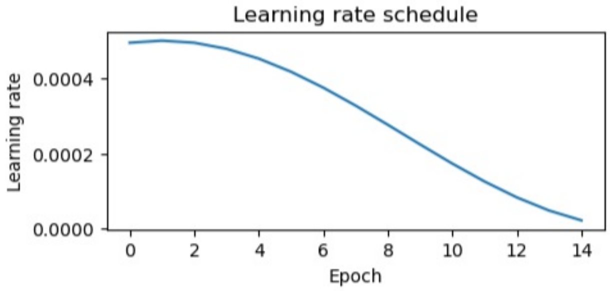


Figure 4.7: A cosine annealing scheduler to adjust learning rate for semantic segmentation model

4.2.2 Fracture detection model

After training the 3D semantic segmentation model to precisely predict the location of the vertebra in the cervical spine, it should be generalized to all the study instances to accurately predict the location of vertebrae in the cervical spine for new study instances that were not included in the training set. By achieving good generalization, the model can be used with confidence to accurately identify and locate fractures in the cervical spine.

In addition, it should be considered that the original view of cervical spine CT scans includes not only the cervical spine but also additional information on other bone parts. Therefore, there is a need to crop the images to the Region of Interest (ROI) to remove the unnecessary parts (Fig 4.8). To accomplish this, YOLOv5 was used to detect vertebrae in the CT scan slices to remove the irrelevant information[24]. YOLOv5 is an object detection algorithm developed by Ultralytics and an evolution of the popular YOLO (You Only Look Once) series of real-time object detection models.

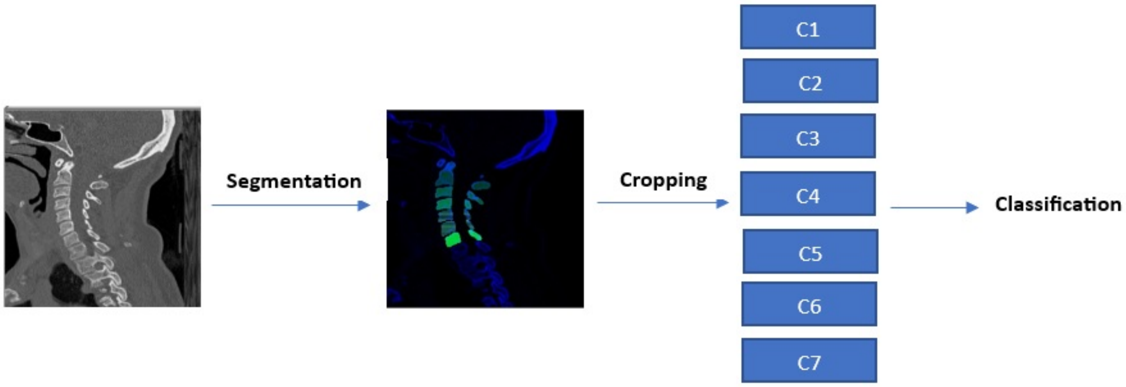


Figure 4.8: The required steps to detect fractures of the cervical spine

In order to crop the cervical spine CT scan slices, the bounding boxes around the vertebrae were detected after generalizing the segmentation model to all the studies and finally the region of interest was cropped for later stage. Also, there is an overlap between cropped voxels of cervical spine CT scans refers to the amount of redundant information that exists between adjacent cropped sections of the scan that may skew the results of the model. This overlap can be minimized by carefully selecting the size and location of each cropped section and by ensuring that there is sufficient spacing between adjacent sections. Therefore, the cropped voxels were resized to the fixed size of 256 × 256 to obtain more accurate results.

Picture 4.9, extracted from the result of the cropping step, shows one slice of a single vertebra and its predicted mask from the segmentation model. We can see that the center part of the vertebrae in the image does not belong to the vertebrae specified by this crop and this is the reason that this part does not appear in the predicted mask.

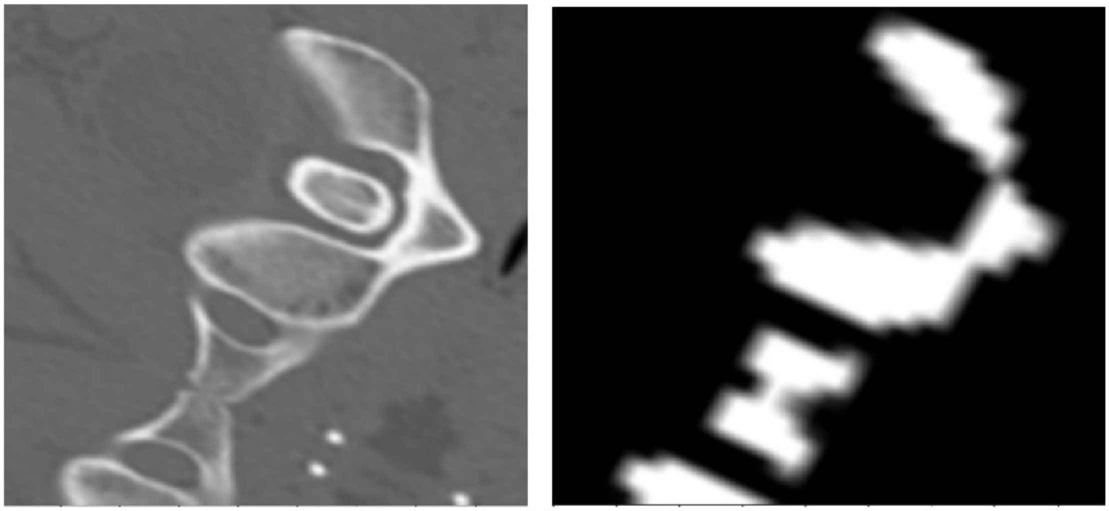


Figure 4.9: From left to right, the first one is the cropped image of one vertebra of the cervical spine related to study instance UID '1.2.826.0.1.3680043.1363' and the second image is its predicted mask by the proposed segmentation mask

Figure 4.10 shows the architecture of the proposed model for the classification problem or fracture detection. It has two steps, which are giving the input slice images to a CNN to extract features of each slice and then using a Bi-GRU layer to predict the value of target. This architecture will predict the probability of a fracture occurring in each cervical vertebra (C1-C7) and then use the sum of them to determine an overall fracture probability for the entire cervical spine. This approach provides a more comprehensive assessment of the risk of cervical spine fractures since it considers the likelihood of multiple vertebrae being affected.

The input to this architecture will be the results of image cropping, seven cervical vertebrae

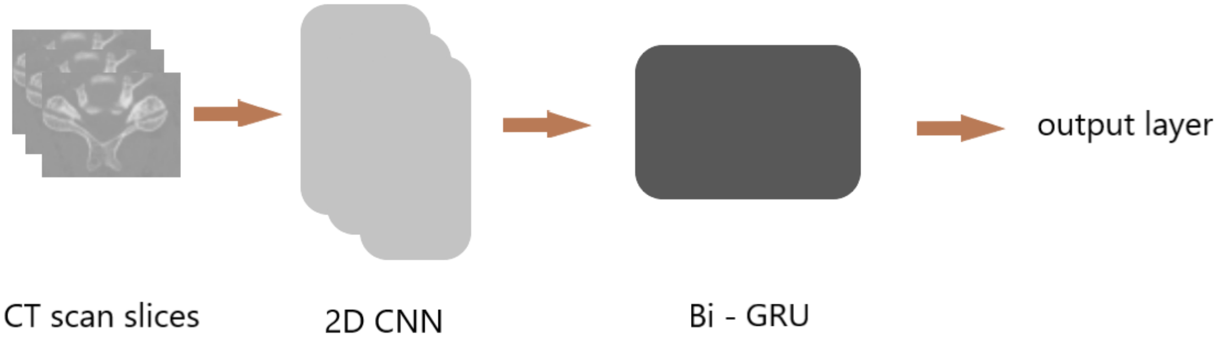


Figure 4.10: CNN + Bi-GRU model architecture to detect fracture in the cervical spine

images for each study instance with a fracture label. Also, for each study instance, the different number of CT scans provided in this dataset and therefore the number of image slices can vary depending on the specific imaging protocol used by the radiology department or imaging center. As a result, for each vertebra, 15 slices were extracted evenly to form the model's input. Therefore, we can represent each image with a vector $\mathrm{x_n}$ and the corresponding image-level label $\mathrm{y_n}$.

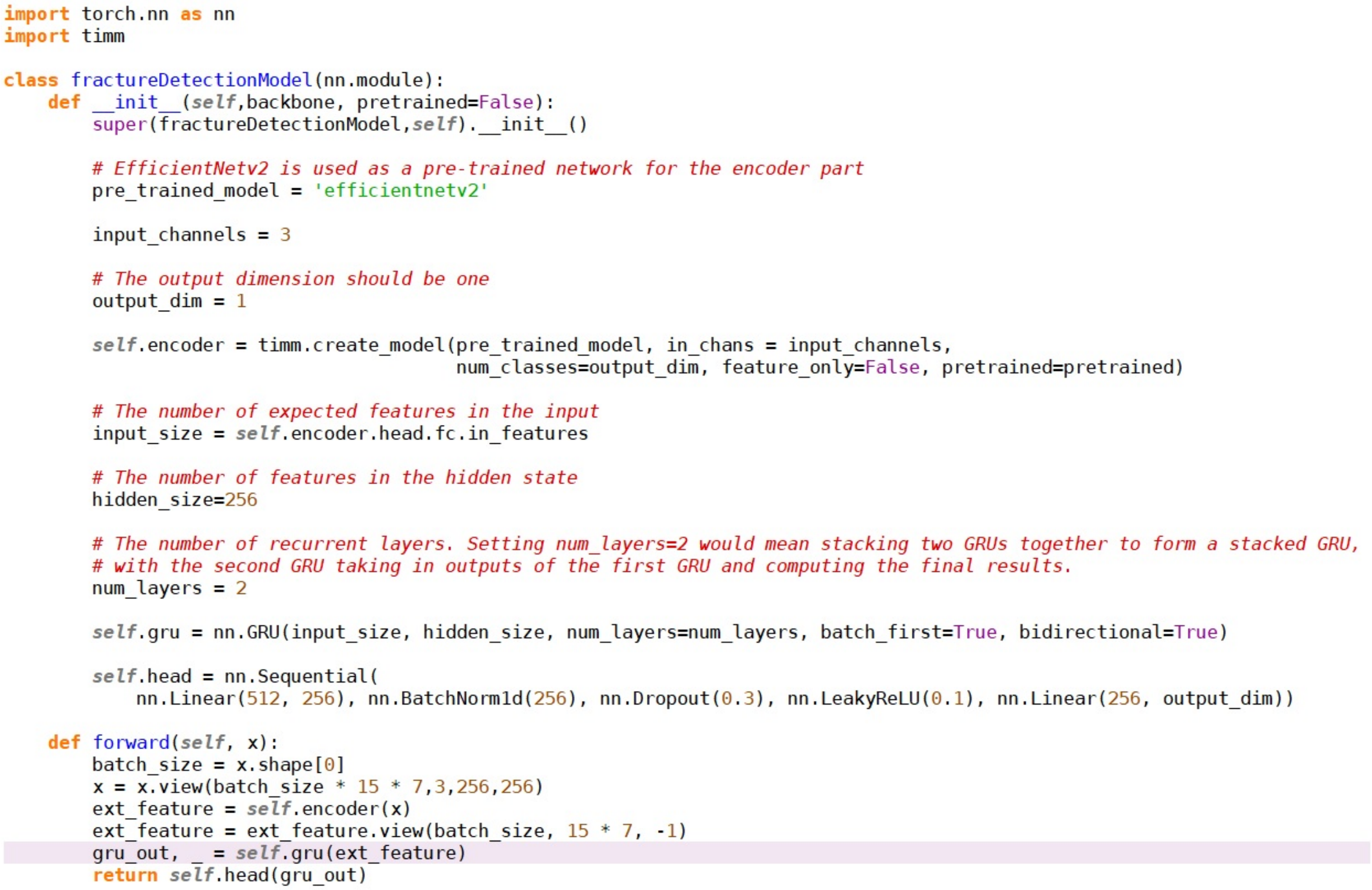
$X = (X_1, X_2, ..., X_{\mathrm{N}})$

$Y = (Y_1, Y_2, ..., Y_{\mathrm{N}})$

where $\mathrm{y_n} = 1$ means the image contains one fracture and $\mathrm{y_n} = 0$ is the opposite.

Similar to the proposed segmentation model, the U-Net architecture was implemented for the classification task and a pre-trained network named EfficientNetv2 was used for the encoder section of the U-Net to ease the training of the network. EfficientNet is a powerful CNN architecture that uses compound coefficients to scale up models in order to be simple and effective. It is also fine-tuned for maximum accuracy but penalized for heavy computational networks. The model also includes batch normalization and ReLU activation functions to improve the efficiency of the training process.

The following pseudo-code shows how the classification model was defined in this research to precisely predict the fracture probability for each vertebra in the given cervical spine CT scan. For training, an AdamW optimizer and a Cosine Annealing scheduler with an initial learning rate of 0.00005 have been used within 5 cross-validation folds of 12 training epochs. Random



```
import torch.nn as nn
import timm

class fractureDetectionModel(nn.module):
    def __init__(self,backbone, pretrained=False):
        super(fractureDetectionModel,self).__init__()

        # EfficientNetv2 is used as a pre-trained network for the encoder part
        pre_trained_model = 'efficientnetv2'

        input_channels = 3

        # The output dimension should be one
        output_dim = 1

        self.encoder = timm.create_model(pre_trained_model, in_chans = input_channels,
                                         num_classes=output_dim, feature_only=False, pretrained=pretrained)

        # The number of expected features in the input
        input_size = self.encoder.head.fc.in_features

        # The number of features in the hidden state
        hidden_size=256

        # The number of recurrent layers. Setting num_layers=2 would mean stacking two GRUs together to form a stacked GRU,
        # with the second GRU taking in outputs of the first GRU and computing the final results.
        num_layers = 2

        self.gru = nn.GRU(input_size, hidden_size, num_layers=num_layers, batch_first=True, bidirectional=True)

        self.head = nn.Sequential(
            nn.Linear(512, 256), nn.BatchNorm1d(256), nn.Dropout(0.3), nn.LeakyReLU(0.1), nn.Linear(256, output_dim))

    def forward(self, x):
        batch_size = x.shape[0]
        x = x.view(batch_size * 15 * 7,3,256,256)
        ext_feature = self.encoder(x)
        ext_feature = ext_feature.view(batch_size, 15 * 7, -1)
        gru_out, _ = self.gru(ext_feature)
        return self.head(gru_out)
```

Figure 4.11: The defined model for the classification stage

resized crop, horizontal flip, and shift scale rotation (50 degrees) have been used as the data augmentation methods. The proposed classification model was trained, evaluated, and tested using a 5-fold cross-validation method. dropout regularization and early stopping methods were applied to prevent overfitting.

Chapter 5

3D Segmentation model

This chapter includes the implementation of a segmentation model to detect the location of the cervical spine in the provided dataset of CT scans. The U-Net architecture was selected to perform segmentation based on the literature review. Fine-tuning of hyperparameters is performed to achieve a reasonable performance to identify vertebra in CT scans. In this chapter, we will compare, evaluate, and discuss the performance of the proposed 3D segmentation model for vertebra detection on CT scans through a 5-fold cross-validation.

5.1 3D U-Net implementation and hyperparameter tunning

The base implementation of the proposed segmentation model was based on the U-Net architecture which the model's implementation was shown in figure 4.5. Most portions of this model involved loading data, doing augmentation, training the model, and an evaluation stage for the model's performance on unseen data.

The figure below shows the architecture of the proposed 3D segmentation model. The model receives 3D CT scans as an input with the shape [batch-size, color-channel, depth, height, width], then the Resnet18d network is used as an encoder to extract the main features which includes multiple convolutional layers, 3 × 3 max-pooling layers followed by 4 layers. The decoder part receives the extracted features and concatenates them for localization information. The segmentation head as a last bock takes input from the decoder and applies a 2D convolution with a 3 × 3 kernel, a stride of (1, 1), and padding of (1,1). The output of this convolutional layer will be in the shape of [batch-size, output-dimension, depth, height, width], which output-dimension corresponds to the number of classes for the segmentation task. The model was trained using a weighted sum of Dice loss and a binary cross-entropy (BCE). This section will explain the details of this architecture.

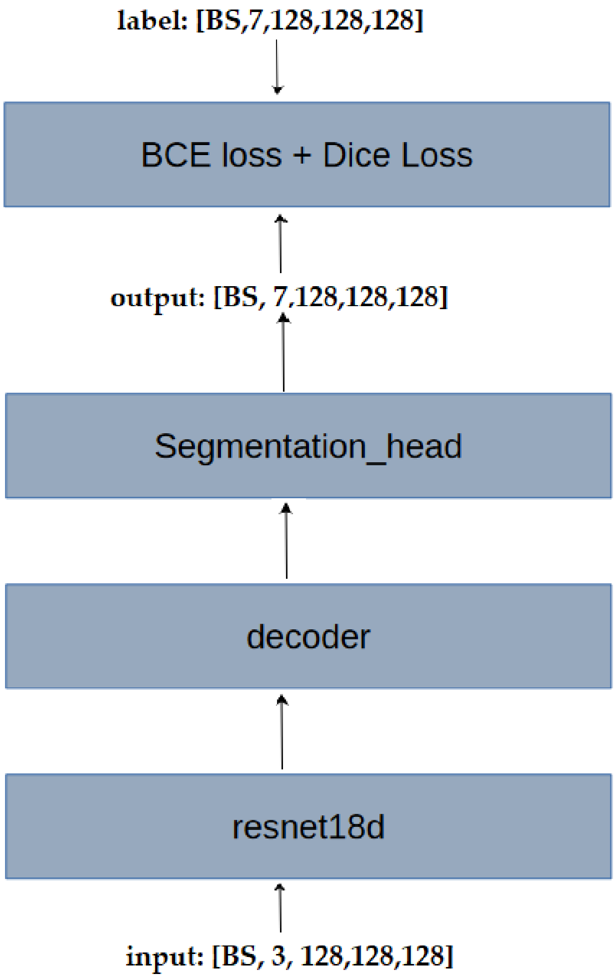

label: [BS,7,128,128,128]
BCE loss + Dice Loss
output: [BS, 7,128,128,128]
Segmentation_head
decoder
resnet18d
input: [BS, 3, 128,128,128]

Figure 5.1: the main architecture of the proposed 3D segmentation model

The initial segmentation model took a long time to choose the best combination of a pre-trained network for the encoder part and the appropriate decoder for the segmentation task because changing the model requires writing a lot of code. The combination of **MONAI** [44] and **segmenation models pytorch** [49], open-source PyTorch frameworks for deep learning in healthcare imaging, was used to improve the process of model training. These libraries streamline the development of deep learning models by offering a set of pre-built components and functions, enabling researchers to focus on their specific research rather than spending time on boilerplate codes.

Therefore, implementing a 3D U-Net model based on **MONAI** for transforming input medical images and **segmenation models pytorch** for the decoder section of U-Net allows us to quickly use different transforms and fine-tune the proposed model to improve the performance of the image segmentation mode.

Using cross-fold validation, 5 folds were generated from the dataset, for each fold, 69 samples as a training set and 18 samples as a validation set were formed (each sample included more than 150 slices of the taken CT scans), and then the score was computed at each CV iteration.

Data augmentation was used to increase the quantity of data and different techniques like Random flipping, affine transformation, and grid distortion were applied in Monai with the 50% chance of applying the flip along the width and height dimension in 3D medical images. The affine transformation includes random translation along each spatial dimension within a dynamically calculated range. The transformation is performed with a 70% probability and uses zero padding to fill the areas outside the transformed image region. These techniques were chosen based on the provided balanced dataset and improvement of the model's performance and generalization.

The hyperparameters such as batch size and optimized epochs were tested and assigned to improve the accuracy of the proposed model. A single 3D CT scan in the provided dataset was $512 \times 512 \times 512$ pixels. Even with the adequate memory resources to process these raw images, the training time was incredibly long for a batch size of 1 without any flexibility to develop the deep learning architecture. Therefore, after resizing the images to $128 \times 128 \times 128$ pixels, different batch sizes including 3 and 4 were tested and finally batch size of 4 was utilized to get a satisfactory result.

During the initial implementation, the value for learning rate was considered 0.01 to control the weights at the end of each batch for which we could not get a satisfactory result. Pytorch has a learning rate scheduler inside the optimizer function that decreases the learning rate during the stochastic gradient descent optimization (SGD) algorithm. Therefore, an initial learning rate of 0.0005 was considered with the CosineAnnealingLR scheduler to automatically adjust the learning rate during training based on its predefined schedules. This scheduler reduced the learning rate from 0.0005 to 0.000001 over time, allowing the model to converge more effectively. Additionally, in order to prevent overfittiong, dropout as a regularization technique was used where during training a fraction of neurons in a layer are randomly set to zero with a defined probability rate. So, the drop_rate parameter in the timm library, an open-source PyTorch-based library, which was used for the encoder section of the proposed U-Net model, was set to 0.01 to reduce the risk of overfitting the training data.

The loss function plays a critical role in the training of the segmentation model and it can significantly affect the performance of the model. The model was trained using a weighted sum of Dice loss and a binary cross-entropy to leverage the flexibility of Dice loss and the curve smoothing property of cross-entropy. The BCEWithLogitsLoss function on PyTorch combines with the defined dice loss function (explained in section 3.2.1) to calculate the dice score for training the segmentation model.

5.2 Experimental Results

Therefore, the first results were collected during the training process over two days by monitoring the dice coefficient losses and F1 scores of training and validation of each fold. Training for each fold was performed for 420 iterations which took about 8 hours on eight RTX A6000 GPU with 48GB memory.

The conclusion revealed a strong segmentation performance for cervical spine regions. Overall, the proposed model achieved an F1 accuracy, a training loss, and a validation loss of around 0.9628, 0.0570, and 0.0288 respectively. Since the dataset was ideally balanced, accuracy was the only metric that mattered. More details on inference performance are listed in Table 1 and visualized in Figure 5.2.

In order to evaluate the model, average evaluation metrics were calculated for each k-fold cross-validation (Table 5.1 and Figure 5.2) to see how well the model can predict the unseen images. These test datasets have been achieved after splitting the main dataset into three sets, training, testing, and validation.

Learning curves provide a mathematical representation of the training process during the repetition tasks. Training and validation losses are two common minimizing metrics that are used to indicate the model performance and a value near to 0.0 shows a perfect training model. On the other hand, accuracy as a maximizing metric is used in the lurving curves meaning that a better score indicates how well the model is learning.

Therefore, Figure 5.2 depicts learning curves of training loss, validation loss, and accuracy versus epoch to compare five folds. In these plots, the shape and dynamics of the learning curves are a way to observe likely underfitting and overfitting in the proposed model. It is obvious that a decreasing trend for the training and validation losses and an increasing trend for accuracy can be seen in these three plots.

we can see that underfitting occurs in the fourth fold and the model could not obtain a sufficiently low validation loss for which the accuracy did not increase anymore. The reason for underfitting in the fourth fold may be that the input data after splitting the dataset for this fold are not a good representative of the whole dataset. Therefore, the fifth fold showed the best performance on almost all evaluation metrics on the data set, while the forth fold cross-validation was worst on the given data set as a result of occurring underfitting, particularly in terms of average validation loss.

No overfitting was observed for the other folds through validation monitoring and as the learning rate decreased the training and validation losses decreased markedly. It is predictable that the accuracy could not significantly increase if training were continued.

Also, Table 5.1 indicates that the third fold achieved better results than the fourth fold mainly because the training dataset for the third fold provided sufficient information to learn the problem.

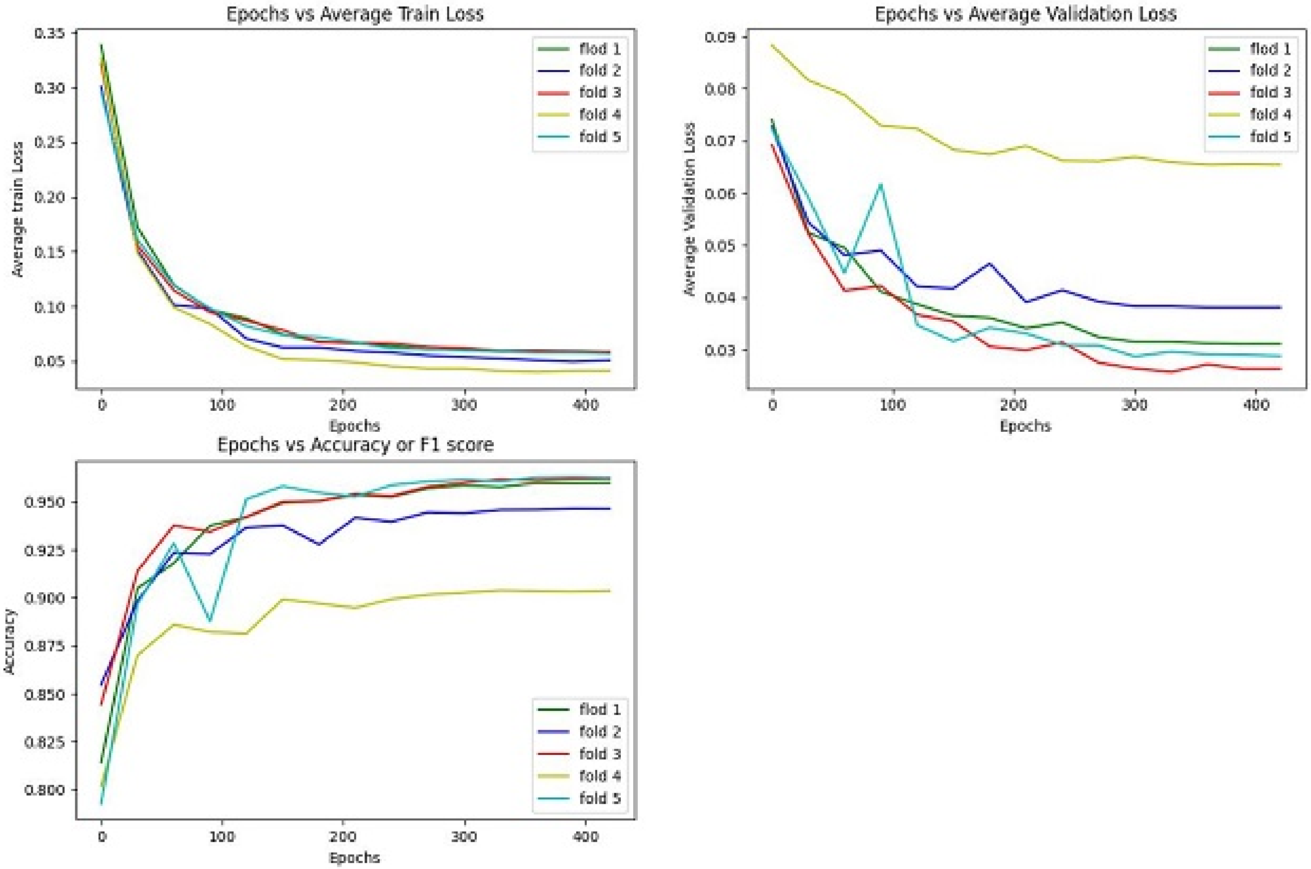
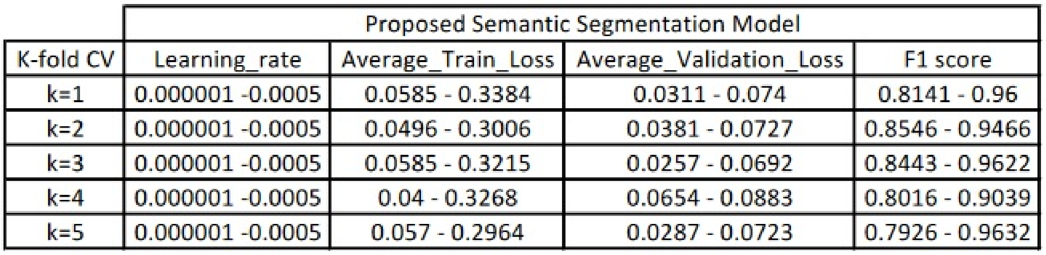


Figure 5.2: Learning curves of five folds



	Proposed Semantic Segmentation Model			
K-fold CV	Learning_rate	Average_Train_Loss	Average_Validation_Loss	F1 score
k=1	0.000001 -0.0005	0.0585 - 0.3384	0.0311 - 0.074	0.8141 - 0.96
k=2	0.000001 -0.0005	0.0496 - 0.3006	0.0381 - 0.0727	0.8546 - 0.9466
k=3	0.000001 -0.0005	0.0585 - 0.3215	0.0257 - 0.0692	0.8443 - 0.9622
k=4	0.000001 -0.0005	0.04 - 0.3268	0.0654 - 0.0883	0.8016 - 0.9039
k=5	0.000001 -0.0005	0.057 - 0.2964	0.0287 - 0.0723	0.7926 - 0.9632

Table 5.1: Achieved results showing the lower and upper bound of the learning rate, average training loss, average validation loss, and F1 score on cervical spine segmentation for each k-fold cross-validation of the studied data set

To show the model performance, we obtained the predicted segmentation mask from the proposed 3D segmentation model with the visualization of the ground truth of the related CT scans. While the masks shown in Figures 5.3 and 4.4 focus on the bone section of the provided CT scans of the cervical vertebrae, they can easily prove that the proposed model correctly distinguished the vertebrae from the provided image.

Figure 5.3 shows one sample of cervical spine CT scans related to C1 to C7 vertebrae with corresponding ground-truth masks and predicted masks from the proposed segmentation model trained on $128 \times 128 \times 128$ image size in the Sagittal plane. The heatmap was generated by the last layer of ResNet18d of the segmentation model. The vertebra regions are in yellow in both ground truth and predicted masks, where the yellow region shows the relevant region. The figure represents the proposed model can correctly do segmentation to detect the location of vertebra in the given CT scan.

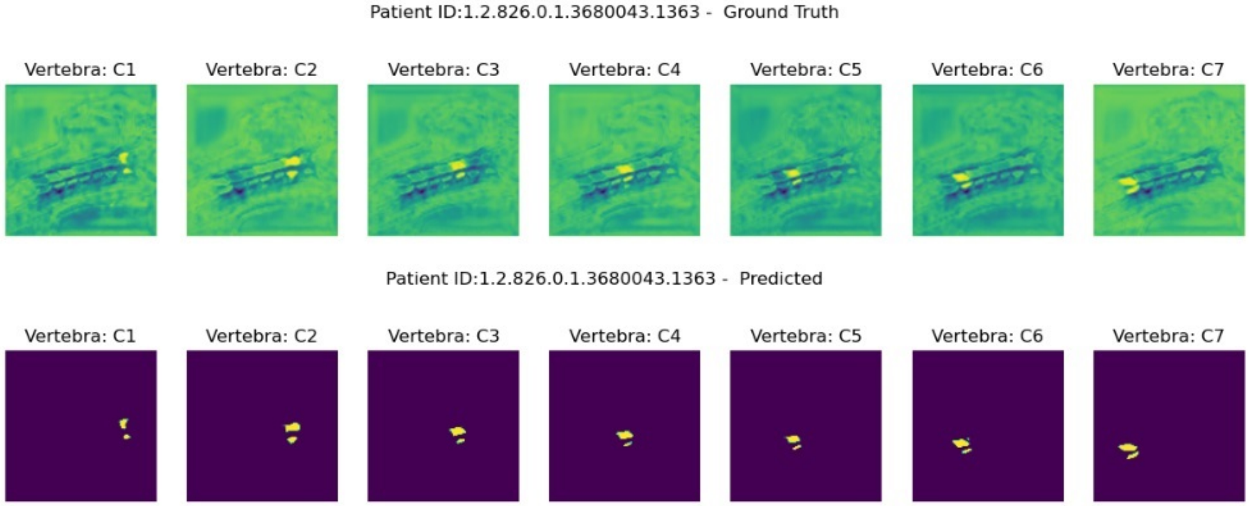
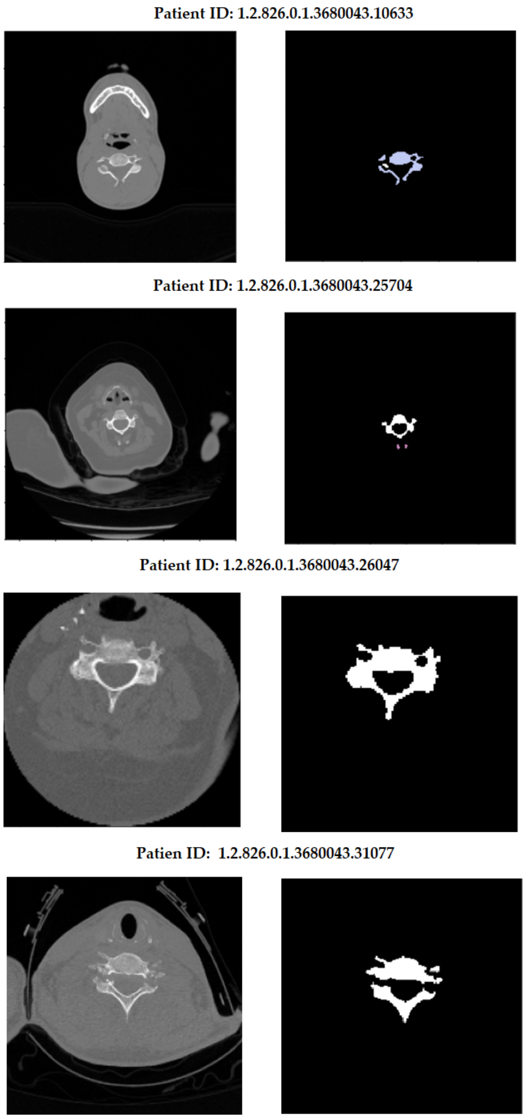


Figure 5.3: Examples of cervical spine CT scans and the predicted segmentation mask from the proposed semantic segmentation model on the validation dataset - sagittal plane



Patient ID: 1.2.826.0.1.3680043.10633

Patient ID: 1.2.826.0.1.3680043.25704

Patient ID: 1.2.826.0.1.3680043.26047

Patien ID: 1.2.826.0.1.3680043.31077

Figure 5.4: predicted segmentation mask for four selected images from the test set in axial view

Figure 5.4 shows our preliminary results of this proposed segmentation model on the cervical spine dataset. While the right figures show the bone section of one vertebra, the left figures depict the CT scans perfectly overlaid with the predicted masks from the U-Net architecture.

5.3 Discussion

One of the biggest challenges for this thesis was the memory constraints. The training images and their corresponding segmentation images in the dataset are around 300GB which needs considerable hard disk capacity, and it needs even more memory to store the cropped CT scan slices. The use of the cloud allows for storing and training of much bigger datasets and would not burden researchers and medical clinics with additional resources to run deep learning models. Also, the strategy of resizing the input size to 128 × 128 × 128, before proceeding to the U-Net segmentation model is more memory efficient and highly advantageous when most 3D models suffer from memory capacity.

Another challenge was training the segmentation and classification models on this big dataset and the need to send the training models and the training dataset to GPU to accelerate operations. Therefore, at first Cedar, a heterogeneous cluster at Simon Fraser University [13], accessible by Compute Canada was used to train the deep learning models. The preliminary results were obtained by training the proposed segmentation model after 20 hours and it only trained for two epochs, after which have achieved 0.81 F1 score. However, unless mini batches were used in the data loader of Torch, an open-source deep learning framework, the process of loading images to GPU was slow and inefficient.

The exploitation of the m7248 datacenter [25], located in Sweden, solved this low speed issue with an interesting observation which was the proposed model trained for around 50 epochs over one hour. Therefore, the m7248 datacenter was used as the computational resource for this research. This data center is comprised of 8 × RTX A6000 graphics processing units with 48 GB GDDR6 of memory each and 16 × AMD EPYC 7252 8-core central processing units with 128 GB of memory.

Most of the previous approaches achieved a high level of accuracy (81-98%) in the detection of cervical spine fractures shown in table 2.1. During the time of writing this thesis a study of evaluating vision transformers (ViT) for detecting cervical spine fractures has been conducted with an accuracy of 98% [18].

However, the performance of the proposed segmentation model was similar to the Convolutional Neural Networks (CNNs) and web-based automatic spine segmentation [36] shown in the previous studies. The achievement of the proposed segmentation mode is significant in the medical industry because the high performance of the U-Net segmentation model performed well in terms of extracted metrics. The encoder and decoder of U-Net enable the extraction of rich features, reducing the time and resources needed to train a model with good performance. Additionally, using a pre-trained network for the encoder section of U-Net is clearly an effective strategy to help the proposed model learn the location of the cervical spine.

It is evident how difficult this segmentation problem is regarding the big size of the training dataset compared to the other datasets used in the previous approaches. However, the visualization in Figure 5.3 suggests that if the model was trained long enough or even deeper pretrained networks were used which needed more computational resources, it would identify the adjacent vertebrae in a single CT scan. Therefore, these improvements could be made to this model to have better performance. For future work, it is more likely to achieve greater accuracy by using more powerful data centers to train the proposed segmentation model.

Also, there are some improvements that could be made to this model such as using different loss functions or different architecture for the encoder and decoder of U-Net in order to get better performance. Due to time and cost limitations, we were unable to perform evaluation metrics on the effects of different loss functions for this dataset which will be the baseline for future works. All of these improvements will be considered in our future research and endeavors.

Chapter 6

Classification model

Once the proposed segmentation model was trained and optimized, it was used to predict the 3D mask for all training images to detect the vertebrae. Then, YOLOv5 was used to crop the regions of interest (ROI) from the cervical spine CT scans. Finally, cropped images with their predicted masks are sent to the classification model to produce fracture probabilities for each vertebra at the cervical spine. This chapter will provide the details related to the required steps to use Ultralytics YOLOv5, the implementation of the proposed classification model, and the results obtained from that model.

This chapter starts by implementing the Yolov5 as an object detection model, and then the result of the predicted bounding box around the vertebra on the three test data. Section 6.2 presents the processing of training data, implementation of data augmentation, and finally the proposed CNN and GRU classification model. Experimental results and evaluation metrics will be discussed in section 6.3.

6.1 Yolov5 implementation

In order to crop the vertebra from the cervical spine CT scans dataset, there is a need to have the coordinates of bounding boxes around the vertebra for all 2019 study instances. The provided dataset includes the coordinates of the bounding boxes around of each vertebra for only

235 study instances. Therefore, Ultralytics YOLOv5 [24] was used to create bounding boxes around the images for which simple text-based files were created to contain the annotations for each image in the dataset. The setup for using Ultralytics YOLOv5 involves the following steps:

- The annotated dataset was converted into the YOLO format, which consists of text files for each image, where each line represents an object in the image with its class label and normalized bounding box coordinates. Each row should have the class index, X_coordinate, Y_coordinate, width and height of the bounding box. As we only have one class (vertebra) a sample .txt file contains [0 0.478515625 0.46484375 0.298828125 0.177734375].

- In Ultralytics YOLOv5, data splitting into training and validation sets are needed to train the model. The split ratio of 80-20 was considered to split data which provided enough data for training while ensuring a sufficient amount for validation.

- A YAML file was created to specify the dataset information, including the path to the training and validation image files, number of classes, and the list of classes. The data configuration YAML file is used during the training process to locate and load the dataset. The sample YAML file created for this problem contains:

  ```
  train: /path/to/train/images
  val: /path/to/val/image
  nc: 1
  names: ['vertebra']
  ```

Finally, the YOLOv5 model was trained on the annotated dataset based on the official YOLOv5 repository's instructions for setting up the training environment and configuring the model [24]. Cloning the repository and installing the requirements have been done by running the following scripts in a Python environment:

```
git clone https://github.com/ultralytics/yolov5
cd yolov5
pip install -r requirements.txt
```

The following training script ran to train the model:

```
python train.py --batch-size 16 --epochs 100 --data /path/to/data.yaml
```

The model was trained for 20,60 and 100 epochs and finally, it has given a satisfactory result with the selection of batch size=16 and epochs=100 with an accuracy of 98%. Then, the trained YOLOv5 model was used to perform inference on all the study instances to generate bounding box predictions for the vertebrae. Figure 6.1 depicts three samples of the cervical spine CT scan and their predicted bounding box around the vertebra with the proposed Yolov5 and predicted segmentation mask. It is evident that the proposed Yolov5 is able to perfectly detect the cervical vertebra from the unseen CT scan.

In conclusion, using YOLOv5 as an object detection model reduced the high amount of irrelevant information and would improve the accuracy of the fracture detection model.

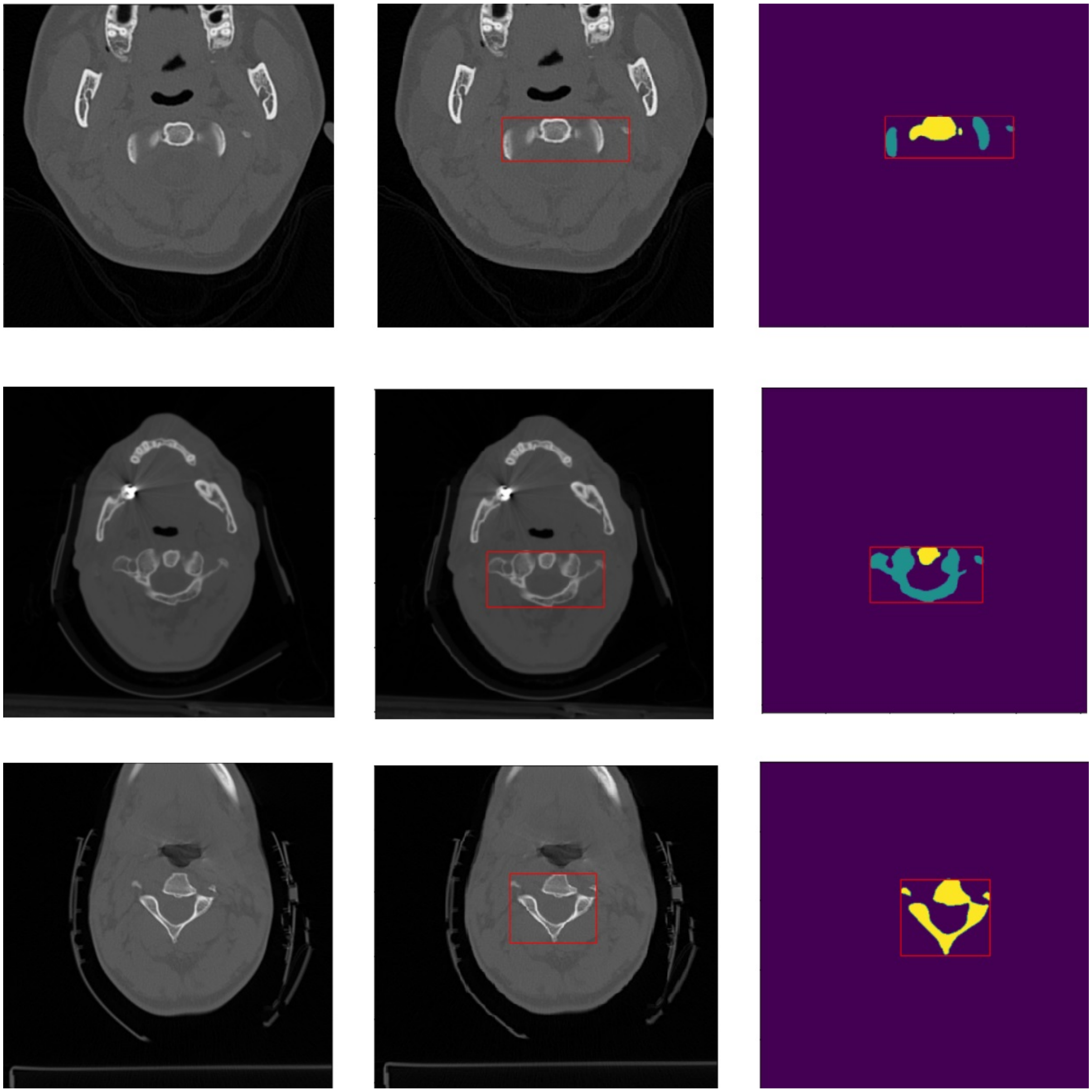


Figure 6.1: Three samples of the predicted bounding box around the vertebra by Yolov5 model - from left to right, the first one is the axial view of one vertebra of the cervical spine, the middle one is the predicted bounding box around that vertebra by Yolov5 and the last one is the predicted bounding box around that vertebra for the related segmentation mask.

6.2 classification

6.2.1 Processing the training data

This large, diverse, and publicly available dataset contains cervical spine CT scans of 2019 cases (961 positive and 1058 negative cases) which corresponds to 711601 images (each case contains an average of 300 slices). Therefore, 30 slices of each vertebra from a case study were extracted to be considered as an input to a 2D CNN. The input shape to the CNN was (batch_size,num_slices, image_size, image_size).

Cropping the CT scans based on the bounding box coordinates, resizing them to 224 × 224, and saving the cropped regions as individual images, produced around 14000 (2019 * 7) samples. Figure 6.2 represents one slice of a cervical vertebra (left image), and its predicted mask (right image) after the cropping process with the Pillow library in Python [48].

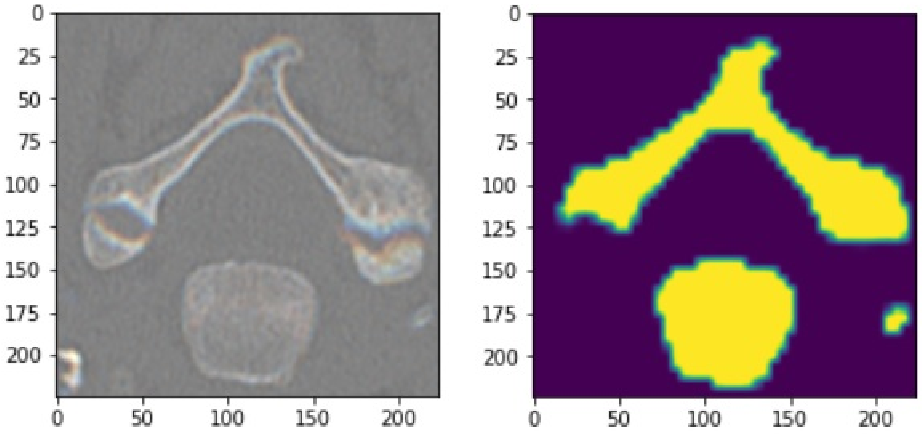


Figure 6.2: one sample of the cropped cervical vertebra CT scan with corresponding mask

Since K-fold cross-validation provides a robust estimate of a model's performance, helps in detecting overfitting and underfitting issues, and allows for a more thorough evaluation of the model's ability to generalize to unseen data, 5-fold cross-validation similar to the segmentation model was used to split the dataset into training, validation, and test subsets.

6.2.2 Implementation of data augmentation

For the augmentation procedure as a crucial part of the training process to prevent overfitting, different augmentation techniques like Resizing, transposing, flipping, and Gaussian blur were used in this classification model to prevent overfitting. These techniques have been chosen based on the problem type and the characteristics of the provided dataset. Also, heaver data augmentation like Grid Distortion and Shift scaling were tested which did not improve the performance of the model. To apply all these image transformations, albumentation, an open-source Python library for image augmentation in deep learning was used to enhance the diversity and quality of the training dataset [2].

Ultimately, a novel data augmentation technique called Mixup was used to improve the accuracy of the proposed model from 96.12% to 96.74%. The mixup technique improves the generalization ability of the deep learning model beyond the traditional empirical risk minimization (ERM) framework [30]. All the augmentation techniques used in the proposed classification model are shown in Table 6.1.

Transforms	Setting	Probability
Resizing	Image size (224,224)	1
HorizontalFlip	Not needed	0.5
VerticalFlip	Not needed	0.5
GuassianBlue	kernel size (0,3)	1
Transpose	Not needed	0.5
Mixup	Not needed	1

Table 6.1: Augmentation techniques and their settings used in the proposed classification model

Traditional ERM methods aim to minimize the average loss over the training data. However, they often struggle with overfitting, especially when the dataset is limited. Mixup addresses this limitation by generating virtual training samples as convex combinations of pairs of original samples and their corresponding labels. This process effectively interpolates between input samples, creating new data points that lie along the line connecting them in the input space. Mixup takes two randomly chosen examples (xi, yi) and (xj, yj) from the training data. It then combines them using the following formulas:

$\bar{X} = \lambda x_{\mathrm{i}} + (1 - \lambda) \times x_{\mathrm{j}}$

$\bar{Y} = \lambda y_{\mathrm{i}} + (1 - \lambda) \times y_{\mathrm{j}}$

Here, λ is a scalar parameter that determines the degree of interpolation and lies between 0 and 1. The resulting virtual example $(\bar{X}, \bar{Y})$ represents a linear combination of the original examples, incorporating the prior knowledge that linear interpolations of input feature vectors should correspond to linear interpolations of their associated labels.

One of the key advantages of mixup is its simplicity and ease of implementation. It requires just a few lines of code and introduces minimal computational overhead compared to traditional data augmentation techniques. Despite its simplicity, mixup consistently improves the generalization performance of deep learning models, as demonstrated by experiments on various image classification tasks.

6.2.3 CNN + GRU Implementation

The following architecture shows an overall view of the proposed classification model, while the pseudo-code of the model is described in Figure 4.11.

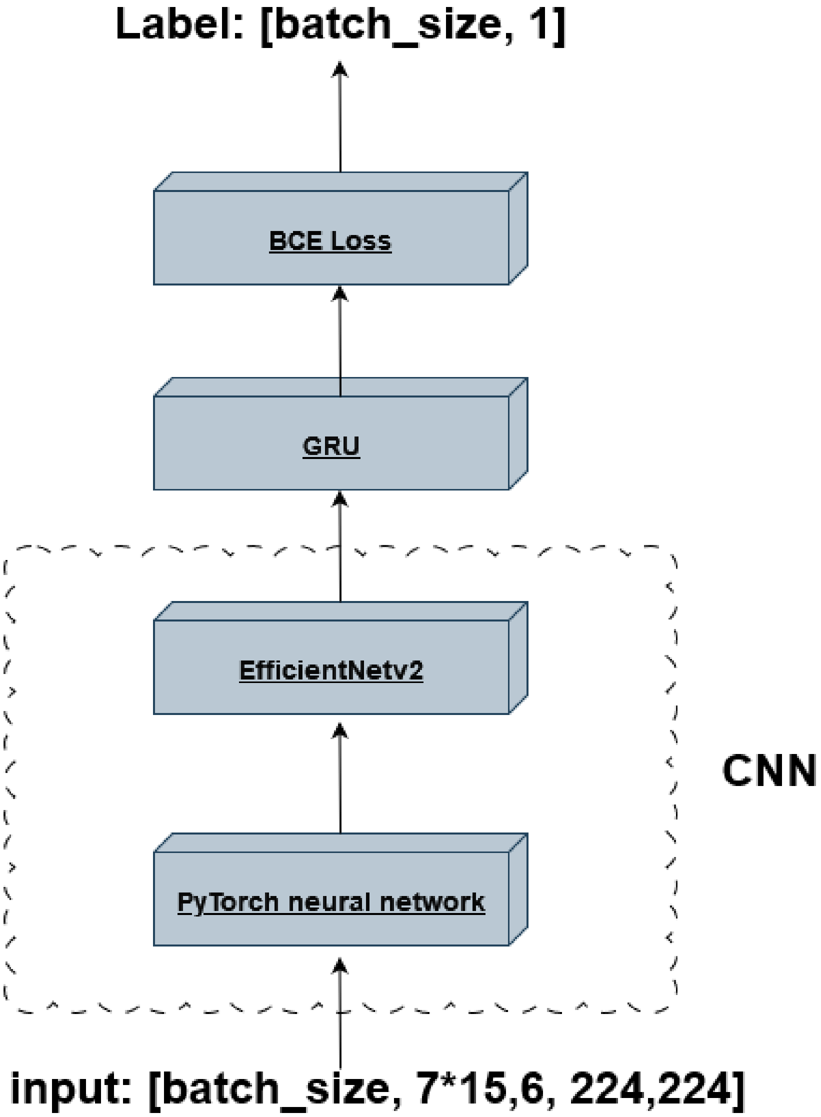


Figure 6.3: An overall view of the proposed classification model

The proposed classification model was defined by sending the input images into a 2D convolutional neural network (CNN) to extract the features from each CT scan and then using a GRU model to create the fracture label. A pre-trained network in PyTorch was chosen for the CNN section.

For selecting the proper backbone for the classification model, ResNet, VGG, and EfficientNet are the popular pre-trained networks for image classification tasks. Also, more complex tasks or datasets may benefit from deeper architectures with more layers and non-linearities, allowing the model to learn intricate features and representations. For problems with limited computational resources, architectures like Resnet and EfficientNet are designed to be lightweight and suitable for deployment on resource-constrained architectures.

Resnet18d and EfficientNet-v2 models were implemented using Timm, an open-source PyTorch library. Initially, ResNet18d was used to learn general features of the cervical spine but it did not work well for the proposed CNN model as a feature extractor. Ultimately, EfficientNet-v2 [43], a pre-trained image classification model, was used to learn the general features of this large dataset.

Moreover, a bi-directional GRU model was used to generate the fracture label for each of the seven cervical vertebrae. The GRU function in the PyTorch library was used to apply a multi-layer gated recurrent unit (GRU) to an input sequence. The number of GRU layers was considered two to form a stacked GRU where the second GRU received the output of the first GRU as an input. The bidirectional parameter in GRU was set to True to get the advantage of Bi-GRU to capture both past and future context, making it more powerful and better suited for the classification task. Although bidirectional models are computationally more expensive and may require more parameters to train, we got a reasonable result by using the m7248 datacenter [25] to train the proposed model.

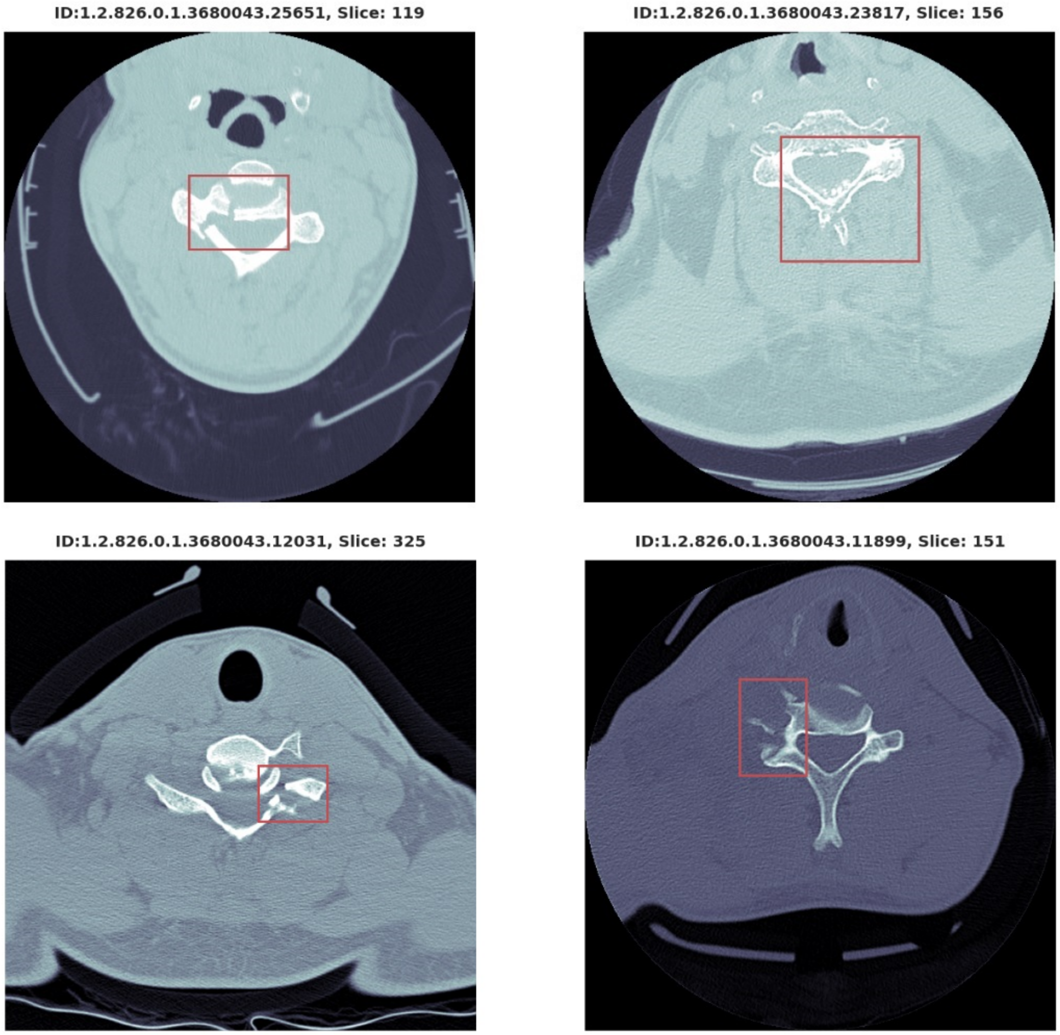
The training setup of the proposed model was as follows: AdamW optimizer with a cosine annealing learning rate, 50 training epochs, batch size of 8, and input size of 224*224.

Finally, the fracture probability for each vertebra was produced by the classification model. The high probability shows that there is mostly likely damage in the given cervical spine CT scan. The damage probability for two study instances has been shown in Table 6.2. It can be seen that the proposed model predicted high fracture probability for C1 and C2 vertebrae related to the first study instance and detected fracture for vertebra C2 in the second study instance.

Ptient ID	Fracture probability
1.2.826.0.1.3680043.6200_C1	0.8211108
1.2.826.0.1.3680043.6200_C2	0.91041774
1.2.826.0.1.3680043.6200_C3	0.04243563
1.2.826.0.1.3680043.6200_C4	0.01370981
1.2.826.0.1.3680043.6200_C5	0.05007572
1.2.826.0.1.3680043.6200_C6	0.21084881
1.2.826.0.1.3680043.6200_C7	0.25250828
1.2.826.0.1.3680043.27262_C1	0.46066386
1.2.826.0.1.3680043.27262_C2	0.89230376
1.2.826.0.1.3680043.27262_C3	0.0756408
1.2.826.0.1.3680043.27262_C4	0.069127046
1.2.826.0.1.3680043.27262_C5	0.022976296
1.2.826.0.1.3680043.27262_C6	0.01965953
1.2.826.0.1.3680043.27262_C7	0.024886766

Table 6.2: Predicting fracture probability for two study instances

Figure 6.4 shows four samples of axial cervical spine images from the unseen test dataset, highlighting the regions where the model focused its attention to predict fracture. The bounding boxes were created around the fractured region in red. From the figure, we can observe that the model can perfectly detect the location of the fracture in the cervical spine CT scans.


ID:1.2.826.0.1.3680043.25651, Slice: 119

ID:1.2.826.0.1.3680043.23817, Slice: 156

ID:1.2.826.0.1.3680043.12031, Slice: 325

ID:1.2.826.0.1.3680043.11899, Slice: 151

Figure 6.4: detected fractures within a single vertebra by the proposed model

6.3 Experimental Results

The classification performance of the proposed model in fracture detection of the cervical spine is presented in Table 6.3. To examine the proposed model's performance and accuracy, the confusion matrix has been used to determine if it is good at the prediction step or not. The accuracy value is 96.74%, the sensitivity value is 78.74% and the specificity value is 98.74%. These results indicate that the proposed model can be used to make predictions. Table 6.3 shows further PPV (Positive predictive value), NPV (Negative Predictive value), and AUC (area under the curve) values.

Metric	TPR	TNR	PPV	NPV	Acc	AUC
result	78.74	98.74	87.48	97.66	96.74	88.74

Table 6.3: Classification performance results on the dataset with 6299 negative cases and 701 positive cases extracted from the proposed model. TPR: sensitivity, TNR: Specificity, PPV: Positive predictive value, NPV: Negative Predictive Value, Acc: Accuracy, AUC: Area under curve. All the values are in percent

The value of 88.74% for the AUC as an important metric to measure the overall performance of the proposed model is considered a good score although it needs to be improved to have a value near or upper than 90%. The TPR value shows that 78.74% of the number of samples belonging to the positive class (fractured) has been classified correctly by the predictive model.

We also observed that the proposed model was unable to correctly detect fractures in some DICOM images. Figure 6.5 shows three samples of false positive error that the model falsely predicted as the fractured vertebrae which are actually intact vertebrae. Any significant differences between the fractures correctly labeled by the proposed model and those they missed were not found after our analysis.

However, since the type of fracture within each slice of the cervical spine has significant importance, it will be considered as future work to categorize all the fractures and examine their effect on the detection of damage and improve the model's performance.

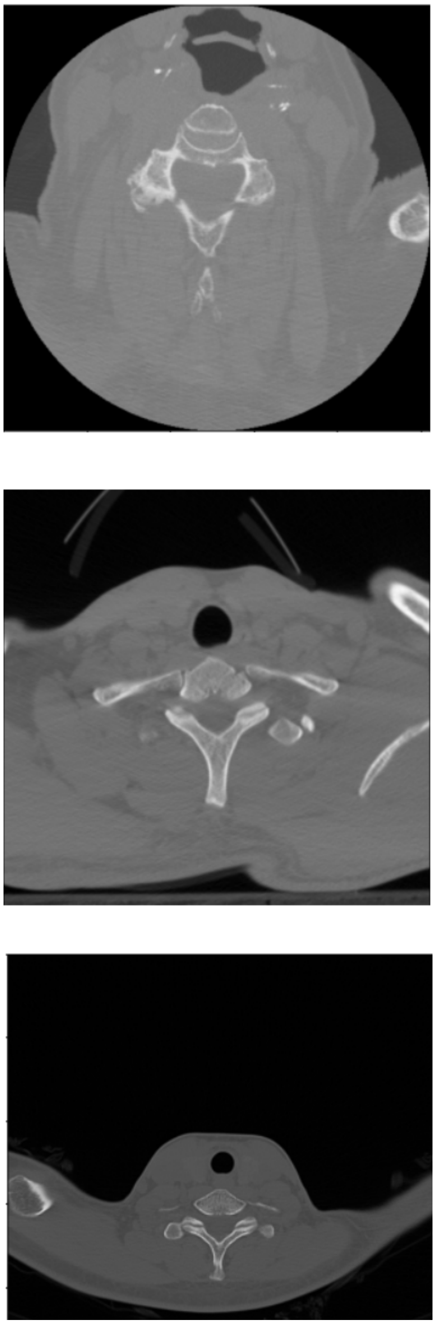


Figure 6.5: flase positive errors in the model related to patient IDs ’1.2.826.0.1.3680043.24606’, ’1.2.826.0.1.3680043.2243’,’1.2.826.0.1.3680043.10005’

6.4 Discussion

In theory, 3D CNN is the most straightforward method for handling this classification problem. Different studies showed that 3D models (an entire volume of the image) are more accurate, represent better performance with limited training data, and are faster to train but require more computational memory compared to 2.5D (five consecutive slices of the image) or 2D (one slice of the image) models [7]. Due to the lack of required GPU memory to train the dataset of this research, it was preferred to use a 2D cnn to predict the presence of fracture in CT scans. However, implementing a 3D classification model will be considered as future work for this research to compare the results with this research.

On the other hand, Our classification model mainly focuses on Convolutional neural networks (CNN) and Residual neural networks to detect the fracture. CNNs excel at recognizing specific features like edges, corners, or textures, but they might struggle to understand how these features interact or contribute to the overall context. Other types of neural networks or approaches might be needed to address this limitation and consider the positioning of features with respect to each other.

Transformers rely heavily on attention mechanisms, which is a concept inspired by human cognitive processes, allowing models to focus on specific parts of input while processing information. Although we used GRU to address this issue in our problem, as a future work, we can use Vision Transformers (ViTs) to bring the powerful self-attention mechanism of Transformers into our classification task.

The EfficientNet-v2 Classification model contains 24 million parameters [43] and requires 5 hours to train the model on the 8 × RTX A6000 GPU with 48GB memory. The whole pipeline took around 7 hours to train the proposed classification model. Different cases have been discussed where the proposed model failed to perform well and we have suggested solutions to overcome them. The proposed classification model has shown that the detection of fractures in cervical spine CT scans can be done automatically by using a deep learning model. This model can be used in the medical industry to help physicians detect fractures in medical images faster, more efficiently, and less expensive.

Chapter 7

Concluding remarks and future works

In this thesis, a new architecture for fracture detection in cervical spine CT images was successfully implemented to minimize the error of the previous studies. In this study, a two steps deep learning model was proposed based on U-Net architecture to detect the location of vertebrae and fracture on 128 ×128×128 image sizes with an accuracy of 96.74%, which demonstrates the capability of deep neural networks to address this challenge. To conclude, this architecture forms a reliable research and has the ability to be applied to critical industries such as medical sciences.

This architecture can be classified into 4 steps. The first step is data preprocessing, loading, and transforming training images in DICOM and NIFTI formats and labeling datasets to training, validation, and testing datasets. The second step is training and validating a segmentation U-Net-based model on the provided cervical spine CT scans. The third step is using that model to predict the segmentation mask for all the study instances and using the Yolov5 model to predict the bounding boxes around all vertebrae to crop them from the DICOM images to reduce the irrelevant information. In the final step, a combination of CNN and GRU models is performed to examine the capability of the model to detect fractures in an unseen medical image.

There are a number of challenges to developing a cervical spine fracture detection model for the provided dataset. One of the primary challenges was the high volume of the training images and the necessity to load them in memory to train the deep-learning models. We solved this issue by using the m7248 datacenter, comprised of 8 × RTX A6000 GPU with 48 GB GDR6 of memory. Therefore, our work demonstrates a pipeline of highly effective 3D segmentation and classification models to detect injury from cervical spine CT scans.

This thesis proposed a deep-learning model to detect the cervical spine fracture in CT scans. The four main contributions of our work are:

Improved the diversity of the training dataset Using mix-up as a novel data augmentation technique produced results that are comparable to the original algorithm.

Improved fracture detection model Adding a step of creating a bounding box around each vertebra to the pipeline, reduced the high amount of irrelevant information, and at the same time, improved the accuracy of the model.

Proposed a flexible architecture This architecture has the flexibility of any changes to the encoder section of U-Net architecture to the state-of-the-art pre-trained networks to make classification over different datasets of CT scan images to detect fractures. This feature results in a unique and flexible framework with acceptable accuracy and robust structure to be applied in highly critical medical applications.

Proposed an integrated cervical spine fracture detection system The above contributions were applied to construct our final integrated system. Experimental results show that our proposed architecture produces better results compared to other popular approaches discussed in the literature review.

In the future, we are planning to use higher input resolution to be trained on the proposed models which will require more powerful computational resources. Our discussion in Chapter 6 prepared the base and foundation for improving the proposed model by including the fracture types in our research and analysis. Moreover, using more depther networks such as ResNet-34 has the advantage of getting higher accuracy or better representation learning which also needs sufficient computational resources to handle the increased model complexity.

Bibliography

[1] A. Criminisi and D. Robertson and E. Konukoglu and J. Shotton and S. Pathak and S. White and K. Siddiqui. “Regression forests for efficient anatomy detection and localization in computed tomography scans”. MA thesis. 2013, pp. 1293–1303. DOI: https://doi.org/10.1016/j.media.2013.01.001. URL: https://www.sciencedirect.com/science/mastersthesis/pii/S1361841513000029.

[2] *Albumentations, a Python library for fast and flexible image augmentations.* en. URL: https://albumentations.ai/.

[3] Alexey Dosovitskiy and Lucas Beyer and Alexander Kolesnikov and Dirk Weissenborn and Xiaohua Zhai and Thomas Unterthiner and Mostafa Dehghani and Matthias Minderer and Georg Heigold and Sylvain Gelly and Jakob Uszkoreit and Neil Houlsby. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. MA thesis. 2020. URL: https://arxiv.org/abs/2010.11929.

[4] Altizer, Linda. “Strains and Sprains”. MA thesis. 2003. URL: https://journals.lww.com/orthopaedicnursing/Fulltext/2003/11000/Strains_and_Sprains.6.aspx.

[5] Alzubaidi, Laith and Zhang, Jinglan and Humaidi, Amjad J. and Al-Dujaili, Ayad and Duan, Ye and Al-Shamma, Omran and Santamaría, J. and Fadhel, Mohammed A. and Al-Amidie, Muthana and Farhan, Laith. “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions”. MA thesis. Mar. 2021, p. 53. DOI: 10.1186/s40537-021-00444-8. URL: https://doi.org/10.1186/s40537-021-00444-8.

[6] Amevo, B and Aprill, C and Bogduk, N. “Abnormal instantaneous axes of rotation in patients with neck pain”. MA thesis. July 1992, pp. 748–756. DOI: 10.1097/00007632-199207000-00004. URL: https://doi.org/10.1097/00007632-199207000-00004.

[7] Arman Avesta and Sajid Hossain and MingDe Lin and Mariam Aboian and Harlan M. Krumholz and Sanjay Aneja. “Comparing 3D, 2.5D, and 2D Approaches to Brain Image Segmentation”. MA thesis. 2022. DOI: 10.1101/2022.11.03.22281923. URL: https://www.medrxiv.org/content/early/2022/11/04/2022.11.03.22281923.

[8] Bali, Akanksha and Singh, Shailendra Narayan. “A Review on the Strategies and Techniques of Image Segmentation”. In: *2015 Fifth International Conference on Advanced Computing Communication Technologies*. 2015, pp. 113–120. DOI: 10.1109/ACCT.2015.63.

[9] Peter J. Bazira. “Clinically applied anatomy of the vertebral column”. MA thesis. 2021, pp. 315–323. DOI: https://doi.org/10.1016/j.mpsur.2021.04.004. URL: https://www.sciencedirect.com/science/mastersthesis/pii/S0263931921000910.

[10] Benjelloun, Mohammed and Mahmoudi. “Spine Localization in X-ray Images Using Interest Point Detection”. MA thesis. June 2009, pp. 309–318. DOI: 10.1007/s10278-007-9099-3. URL: https://doi.org/10.1007/s10278-007-9099-3.

[11] Richard Bibb. “3 - Export data format and media”. In: *Medical Modelling*. Ed. by Richard Bibb. Woodhead Publishing Series in Biomaterials. Woodhead Publishing, 2006, pp. 32–36. ISBN: 978-1-84569-138-7. DOI: https://doi.org/10.1533/9781845692001.32. URL: https://www.sciencedirect.com/science/mastersthesis/pii/B9781845691387500034.

[12] Carolyn Perry MSc, Francesca Salvador MSc. URL: https://www.kenhub.com/en/library/anatomy/cervical-spine.

[13] *Cedar, a heterogeneous cluster*. en. URL: https://docs.alliancecan.ca/wiki/Cedar.

[14] *Cervical Spondylosis (Arthritis of the Neck) - OrthoInfo - AAOS*. en. URL: https://www.orthoinfo.org/en/diseases--conditions/cervical-spondylosis-arthritis-of-the-neck/.

[15] Chapman Hall/CRC. *Machine Learning an algorithmic perspective*. 2015.

[16] Charu C. Aggarwal. *Neural Networks and deep learning*. 2018.

[17] Chen, J. X. and Jiang, D. M. and Zhang, Y. N. "A Hierarchical Bidirectional GRU Model With Attention for EEG-Based Emotion Classification". MA thesis. 2019, pp. 118530–118540. DOI: 10.1109/ACCESS.2019.2936817.

[18] Chład, Paweł and Ogiela, Marek R. "Deep Learning and Cloud-Based Computation for Cervical Spine Fracture Detection System". MA thesis. 2023. DOI: 10.3390/electronics12092056. URL: https://www.mdpi.com/2079-9292/12/9/2056.

[19] DeSai, Charisma and Reddy, Vamsi and Agarwal, Amit. *Anatomy, Back, Vertebral Column*. StatPearls Publishing, Treasure Island (FL), 2022. URL: http://europepmc.org/books/NBK525969.

[20] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1412.6980.

[21] Fenster, A. and Chiu, B. "Evaluation of Segmentation Algorithms for Medical Imaging". In: *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. 2005, pp. 7186–7189. DOI: 10.1109/IEMBS.2005.1616166.

[22] Flanders. *RSNA 2022 Cervical Spine Fracture Detection*. 2022. URL: https://www.kaggle.com/competitions/rsna-2022-cervical-spine-fracture-detection/overview.

[23] Ghojogh, Benyamin and Crowley, Mark. *The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial.* 2019. DOI: `10.48550/ARXIV.1905.12787`. URL: `https://arxiv.org/abs/1905.12787`.

[24] Glenn Jocher1, Ayush Chaurasia, Alex Stoken2, Jirka Borovec3. *ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation.* Version v7.0. Nov. 2022. DOI: `10.5281/zenodo.7347926`. URL: `https://doi.org/10.5281/zenodo.7347926`.

[25] *Global GPU market.* en. URL: `https://cloud.vast.ai/create/`.

[26] Glocker, Ben and Feulner, J and Criminisi, Antonio and Haynor, D R and Konukoglu, E. "Automatic localization and identification of vertebrae in arbitrary field-of-view CT scans". en. MA thesis. Germany, 2012, pp. 590–598.

[27] Grebe, Hank. *Lateral View Of A Woman's Skull And Cervical Spine.* en. URL: `https://www.greatbigcanvas.com/view/lateral-view-of-a-womans-skull-and-cervical-spine-labeled,2595590/`.

[28] Hojjat Salehinejad, Joseph Barfett, Shahrokh Valaee, Timothy Dowdell. "Cylindrical Transform: 3D Semantic Segmentation of Kidneys With Limited Annotated Images". MA thesis. 2018. URL: `http://arxiv.org/abs/1809.10245`.

[29] Hojjat Salehinejad, Joseph Barfett, Shahrokh Valaee, Timothy Dowdell. "Training Neural Networks with Very Little Data - A Draft". MA thesis. 2017. URL: `http://arxiv.org/abs/1708.04347`.

[30] Hongyi Zhang and Moustapha Cissé and Yann N. Dauphin and David Lopez-Paz. "mixup: Beyond Empirical Risk Minimization". MA thesis. 2017. URL: `http://arxiv.org/abs/1710.09412`.

[31] Ilya Loshchilov and Frank Hutter. "Fixing Weight Decay Regularization in Adam". MA thesis. 2017. URL: `http://arxiv.org/abs/1711.05101`.

[32] Ilya Loshchilov and Frank Hutter. "SGDR: Stochastic Gradient Descent with Restarts". MA thesis. 2016. URL: `http://arxiv.org/abs/1608.03983`.

[33] Jadon, Shruti. “A survey of loss functions for semantic segmentation”. In: *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. 2020, pp. 1–7. DOI: 10.1109/CIBCB48159.2020.9277638.

[34] Kadom, Nadja and Khademian, Zarir and Vezina, Gilbert and Shalaby-Rana, Eglal and Rice, Amy and Hinds, Tanya. “Usefulness of MRI detection of cervical spine and brain injuries in the evaluation of abusive head trauma”. MA thesis. July 2014, pp. 839–848.

[35] Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun. “Deep Residual Learning for Image Recognition”. MA thesis. 2015. URL: http://arxiv.org/abs/1512.03385.

[36] Kim Young Jae,Ganbold Bilegt,Kim Kwang Gi. “Web-Based Spine Segmentation Using Deep Learning in Computed Tomography Images”. In: *Healthc Inform Res* 26.1 (2020), pp. 61–67. DOI: 10.4258/hir.2020.26.1.61. URL: http://e-hir.org/journal/view.php?number=1015.

[37] KyungHyun Cho and Bart van Merrienboer and Dzmitry Bahdanau and Yoshua Bengio. “On the Properties of Neural Machine Translation: Encoder-Decoder Approaches”. MA thesis. 2014. URL: http://arxiv.org/abs/1409.1259.

[38] Lee, Christoph I. and Haims, Andrew H. and Monico, Edward P. and Brink, James A. and Forman, Howard P. “Diagnostic CT Scans: Assessment of Patient, Physician, and Radiologist Awareness of Radiation Dose and Possible Risks”. In: *Radiology* 231.2 (2004). PMID: 15031431, pp. 393–398. DOI: 10.1148/radiol.2312030767. URL: https://doi.org/10.1148/radiol.2312030767.

[39] Long, L.R. and Thoma, G.R. “Use of shape models to search digitized spine X-rays”. In: *Proceedings 13th IEEE Symposium on Computer-Based Medical Systems. CBMS 2000.* 2000, pp. 255–260. DOI: 10.1109/CBMS.2000.856908.

[40] Luis Perez and Jason Wang. “The Effectiveness of Data Augmentation in Image Classification using Deep Learning”. MA thesis. 2017. URL: http://arxiv.org/abs/1712.04621.

[41] Ma Yi-de and Liu Qing and Qian Zhi-bai. “Automated image segmentation using improved PCNN model based on cross-entropy”. In: *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.* 2004, pp. 743–746. DOI: `10.1109/ISIMP.2004.1434171`.

[42] Maury R. Ellenberg and Joseph C. Honet and Walter J. Treanor. “Cervical radiculopathy”. MA thesis. 1994, pp. 342–352. DOI: `https://doi.org/10.1016/0003-9993(94)90040-X`. URL: `https://www.sciencedirect.com/science/mastersthesis/pii/000399939490040X`.

[43] Mingxing Tan and Quoc V. Le. “EfficientNetV2: Smaller Models and Faster Training”. MA thesis. 2021. URL: `https://arxiv.org/abs/2104.00298`.

[44] *Monai, PyTorch framework.* en. URL: `https://catalog.ngc.nvidia.com/orgs/nvidia/teams/clara/containers/monai-toolkit`.

[45] Mower, W R and Hoffman, J R and Pollack, Jr, C V and Zucker, M I and Browne, B J. “Use of plain radiography to screen for cervical spine injuries”. en. MA thesis. July 2001, pp. 1–7.

[46] Olaf Ronneberger and Philipp Fischer and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. MA thesis. 2015. URL: `http://arxiv.org/abs/1505.04597`.

[47] Nobuyuki Otsu. “A threshold selection method from gray-level histograms”. MA thesis. 1979, pp. 62–66.

[48] *Pillow, a Python library for image manipulation.*

[49] *Python library with Neural Networks for Image Segmentation based on PyTorch.* en. URL: `https://github.com/qubvel/segmentation_models.pytorch`.

[50] R. Forghani and S.K. Mukherji. “Advanced dual-energy CT applications for the evaluation of the soft tissues of the neck”. Special Focus Edition on Head and Neck Imaging. MA thesis. 2018, pp. 70–80. DOI: https://doi.org/10.1016/j.crad.2017.04.002. URL: https://www.sciencedirect.com/science/mastersthesis/pii/S0009926017301393.

[51] Rahman, Md Atiqur and Wang, Yang. “Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation”. In: *Advances in Visual Computing*. Ed. by George Bebis et al. Cham: Springer International Publishing, 2016, pp. 234–244. ISBN: 978-3-319-50835-1.

[52] *Responding to a Cervical Spinal Cord Injury - InSync Physiotherapy*. en. URL: https://insyncphysio.com/responding-to-a-cervical-spinal-cord-injury/.

[53] Robert W. Jahnke and Blaine L. Hart. “Cervical Stenosis, Spondylosis, and Herniated Disc Disease”. MA thesis. 1991, pp. 777–791. DOI: https://doi.org/10.1016/S0033-8389(22)02083-8. URL: https://www.sciencedirect.com/science/mastersthesis/pii/S0033838922020838.

[54] Roy, Payel and Dutta, Saurab and Dey, Nilanjan and Dey, Goutami and Chakraborty, Sayan and Ray, Ruben. “Adaptive thresholding: A comparative study”. In: *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*. 2014, pp. 1182–1186. DOI: 10.1109/ICCICCT.2014.6993140.

[55] S.M. Masudur Rahman, Muhammad Asad, Michael Gundry, Karen Knapp and Greg Slabaugh. “Patch-based corner detection for cervical vertebrae in X-ray images”. MA thesis. 2017, pp. 27–36. DOI: https://doi.org/10.1016/j.image.2017.04.002. URL: https://www.sciencedirect.com/science/mastersthesis/pii/S0923596517300681.

[56] Saleem and Balaj. *Medical image formats: An introduction*. Sept. 2022. URL: https://ango.ai/medical-image-formats-introduction/.

[57] Sang Uk Lee and Seok Yoon Chung and Rae Hong Park. “A comparative performance study of several global thresholding techniques for segmentation”. MA thesis. 1990,

pp. 171–190. DOI: https://doi.org/10.1016/0734-189X(90)90053-X. URL: https://www.sciencedirect.com/science/mastersthesis/pii/0734189X9090053X.

[58] Seff, Ari and Lu, Le and Barbu, Adrian and Roth, Holger and Shin, Hoo-Chang and Summers, Ronald M. “Leveraging Mid-Level Semantic Boundary Cues for Automated Lymph Node Detection”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 53–61.

[59] Shervin Minaee and Yuri Boykov and Fatih Porikli and Antonio Plaza and Nasser Kehtarnavaz and Demetri Terzopoulos. “Image Segmentation Using Deep Learning: A Survey”. MA thesis. 2020. URL: https://arxiv.org/abs/2001.05566.

[60] Small, J.E. and Osler, P. and Paul, A.B. and Kunst, M. “CT Cervical Spine Fracture Detection Using a Convolutional Neural Network”. MA thesis. 2021, pp. 1341–1347. DOI: 10.3174/ajnr.A7094. URL: http://www.ajnr.org/content/42/7/1341.

[61] *structure of cervical spine*. en. URL: https://musculoskeletalkey.com/cervical-spine-10/.

[62] Swartz, Erik E and Floyd, R T and Cendoma, Mike. “Cervical spine functional anatomy and the biomechanics of injury due to compressive loading”. en. MA thesis. United States, July 2005, pp. 155–161.

[63] Tanaka, Nobuhiro and Atesok, Kivanc and Nakanishi, Kazuyoshi and Kamei, Naosuke and Nakamae, Toshio and Kotaka, Shinji and Adachi, Nobuo. “Pathology and Treatment of Traumatic Cervical Spine Syndrome: Whiplash Injury”. MA thesis. Feb. 2018, p. 4765050. DOI: 10.1155/2018/4765050. URL: https://doi.org/10.1155/2018/4765050.

[64] Van Eerd, Maarten and Patijn, Jacob and Sieben, Judith M. and Sommer, Mischa and Van Zundert, Jan and van Kleef, Maarten and Lataster, Arno. “Ultrasonography of the Cervical Spine: An In Vitro Anatomical Validation Model”. MA thesis. Jan. 2014,

pp. 86–96. DOI: 10.1097/ALN.0000000000000006. URL: https://doi.org/10.1097/ALN.0000000000000006.

[65] *What to Know About Spinal Ligaments*. en. URL: https://www.healthcentral.com/condition/back-pain/ligaments.

[66] Zheng, Zhenyu and Chen, Zhencheng and Hu, Fangrong and Zhu, Jianming and Tang, Qunfeng and Liang, Yongbo. "An Automatic Diagnosis of Arrhythmias Using a Combination of CNN and LSTM Technology". MA thesis. 2020. DOI: 10.3390/electronics9010121. URL: https://www.mdpi.com/2079-9292/9/1/121.

[67] Zhou, Zhiyu and Gao, Manman and Wei, Fuxin and Liang, Jiabi and Deng, Wenbin and Dai, Xuejun and Zhou, Guangqian and Zou, Xuenong. "Shock Absorbing Function Study on Denucleated Intervertebral Disc with or without Hydrogel Injection through Static and Dynamic Biomechanical Tests In Vitro". MA thesis. June 2014, p. 461724. DOI: 10.1155/2014/461724. URL: https://doi.org/10.1155/2014/461724.