Accelerating Genetic Screening of Skeletal Conditions using Zebrafish and AI

Katie Noonan

A thesis submitted to University College Dublin
in fulfilment of the requirements for the degree of

ME in Biomedical Engineering

College of Engineering and Architecture
School of Electronic & Electrical Engineering

Head of School: Professor Peter Kennedy
Supervisors: Dr. Kathleen Curran, Dr. John Healy

30 April 2021

STATEMENT OF ORIGINAL AUTHORSHIP

I hereby certify that the submitted work is my own work, was completed while registered as a candidate for the degree stated on the Title Page, and I have not obtained a degree elsewhere on the basis of the research presented in this submitted work.

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank all the people who have supported me over the past years through my pursuit of achieving my Master's in Biomedical Engineering. In this regard, I would like to express my immense gratitude to my friends and family that have provided such a solid supporting framework that has continuously pushed me through many challenges that I have faced over the years.

I would like to deeply thank my supervisors, both Dr. Kathleen Curran and Dr. John Healy, for their continuous support and guidance including every ounce of enthusiastic encouragement and constructive critiques over the entire course of this study.

I would also like to recognise the invaluable guidance and support provided by the Zebrafish Team which include Dr. Erika Kague, Yushi Yang and Adil Dahlan. The work outlined in this project would not have been possible or done to its standard without their input and constant collaboration.

All this, for a fish? - Graeme Noonan

CONTENTS

ABSTRACT

Osteoporosis is a health condition that weakens bone due to diminished bone mineral densities and can be evaluated using X-rays. This bone disorder arises as a result of genetic mutations being expressed during bone development. Previous research studies indicate over 500 genetic loci have been associated with this disease. Functional studies need to be performed to test associated genetic loci and evaluate its resulting phenotype in order for research communities to identify therapeutic targets. By combining the task-solving abilities of Artificial Intelligence with the statistical power of zebrafish, genetic screening can be accelerated leading to rapid genetic discoveries. Many publications involve the automatic segmentation and analysis of the human vertebral column but previous attempts in zebrafish have been limitied to the quantification of spine curvature using semi-automated vertebrae segmentation frameworks. This project outlines the development of two completely novel and different ensemble segmentation models, composed of four constituent segmentation models, specifically designed for the zebrafish skeleton using 812 annotated and validated zebrafish X-ray samples. The spine segmentation ensemble achieved an average Dice Similarity Coefficient of 0.92, Precision of 0.96 and Sensitivity of 0.89. The vertebrae segmentation ensemble achieved an average Dice Similarity Coefficient of 0.79, Precision of 0.77 and Sensitivity of 0.80. This study also outlines four fully automatic and highly innovative frameworks implemented for the rapid analysis of zebrafish spine length and curvature as well as vertebra length and area. Further to this, these automatic frameworks were statistically indifferent to that of the manual measurements performed by a leading geneticist and zebrafish expert.

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AI Artificial Intelligence.

AP Anterior-Posterior.

ASM Active Shape Model.

BO Binary Output.

CaHA Calcium Hydroxyapatite phantom.

CNN Convolutional Neural Network.

CPU Central Processing Unit.

DL Deep Learning.

DU-Net Deformable U-Net.

FN False Negative.

FP False Positive.

GPU Graphics Processing Unit.

Grad-CAM Gradient-weighted Class Activation Mapping.

GT Ground Truth.

GUI Graphical User Interface.

IDF Invention Disclosure Form.

ISBI International Symposium on Biomedical Imaging.

LAT Lateral.

ML Machine Learning.

MVC-Net Multi-View Correlation Network.

MVE-Net Multi-View Extrapolation Net.

NO Network Output.

ReLU Rectified Linear Unit.

ROI Region of Interest.

TN True Negative.

TP True Positive.

XAI Explainable Artificial Intelligence.

GLOSSARY OF STATISTICAL METRICS

Balanced Accuracy Rate An accuracy metric that accounts for class imbalance by taking the proportion of each class into account.

Confidence Interval A statistical interval describing a set range of values surrounding the sample mean based on the observed data. The probability in which a sample point lies within the range depends on the confidence level.

Confusion Matrix A table that summarises the prediction results of a classification problem.

Dice Similarity Coefficient A statistical measure of the overlap between two sets of data.

F_β Score A metric that evaluates the weighted harmonic mean of precision and sensitivity.

Mean Absolute Error An evaluation metric used to measure the the average absolute values of individual prediction errors over all instances of a test set.

Pearson Correlation Coefficient A statistical measure of the linear correlation between two sets of data.

Precision The proportion of positive predicted samples that are truly positive.

Sensitivity The proportion of truly positive samples that are predicted positive.

Student T-test A statistical test used to determine the significance of the differences between two datasets.

CHAPTER

ONE

INTRODUCTION

Osteogenesis imperfecta, scoliosis and osteoporosis are a tiny sample of bone disorders that arise as a result of gene mutations playing a role in bone development. Previous studies have identified several genomic regions associated with skeletal conditions. For example, over 500 genetic loci have been associated with the development of osteoporosis and another 100 have been associated with osteoarthritis. However, the location of the gene on the chromosome does not directly associate with being the direct cause of the skeletal condition as one genetic loci could hold several genes. As a result, functional studies are needed to identify the specific genes responsible for particular conditions. Identification of causal genes allows research communities to find therapeutical targets for specific diseases through translational genomics. However, given the shear number of genes that would need to be functionally tested, there is a necessity for faster screening processes to allow for rapid genetic discoveries.

Functional studies or mutagenesis cannot be carried out on humans due to legal and ethical implications. Human growth is also considerably slow for phenotypic screening of genetic mutations too. This is overcome by employing the use of animal models that exhibit similar genomic structures and physiology. Rodents are often used as animal models. However, testing a large number of genes in rodents by generating knockout animals has proven to be costly and

time-consuming. This study is centred around the use of the zebrafish for genetic screening of skeletal conditions.

Within the medical field, the use of Artificial Intelligence (AI) has gained traction over previous years. This is as a direct result of the advancements of both computer vision and computational power that can efficiently support the development of large Deep Learning (DL) networks. One example of its many applications includes the automation of segmentation tasks, whereby an image is separated into different regions due to the expression of similar properties. With regards to the skeleton, previous segmentation publications have mainly focused on the vertebral column in humans, with very little extending further on to their individual vertebrae and/or other animals. AI is yet to be applied to the assessment of skeletal conditions using fish models to the same caliber as seen in humans.

This study was performed in collaboration with Dr. Erika Kague, a leading geneticist and zebrafish expert in the UK, as well as her colleagues. Dr. Kague is specifically interested in identifying genetic factors involved in the development of osteoporosis through the use of zebrafish as animal models. This involves the rapid generation of zebrafish with mutations in candidate genes for osteoporosis and looking for vertebral column changes using X-rays.

1.1 Overall Aim

This study aims to automate the analysis of zebrafish X-rays to allow for in vivo rapid screening of zebrafish carrying mutations in putative candidate genes in skeletal conditions.

1.2 Specific Objectives

The specific objectives of this study, as developed in collaboration with Dr. Kague, are as follows:

1. To design a reliable pipeline that provides meaningful analyses of the zebrafish spine. This constitutes of the following tasks:

    - Development of an accurate and reliable spinal segmentation model for zebrafish X-rays.
    - Development of a framework that can automatically measure the length of the spine, to a similar standard of a domain expert, based on the results of the spinal segmentation model.
    - Development of a framework that can automatically quantify the spinal curvature based on the results of the spinal segmentation model.

2. To design a reliable pipeline that provides meaningful analyses of the zebrafish vertebrae. This constitutes of the following tasks:

    - Development of an accurate and reliable vertebrae segmentation model for zebrafish X-rays.
    - Development of a framework that can automatically measure the length of each vertebra identified in the output of the vertebrae segmentation model, to a similar standard of a domain expert.
    - Development of a framework that can automatically measure the area of each vertebra identified in the output of the vertebrae segmentation model, to a similar standard of a domain expert.

1.3 Novel Contribution

This study outlines the very first automated pipelines for rapid analysis of the zebrafish skeleton using X-rays. Through the successful completion of this project, two ensemble models - composed of four segmentation neural network models - were specifically designed for the analysis of zebrafish phenotypes in conjunction with regular collaboration with Dr. Kague. This framework will allow in-vivo screening of a large number of zebrafish in a shorter time-frame, further supporting in-vivo validation of genetic factors involved in bone diseases.

1.4 Thesis Layout

This thesis will begin with a literature review, consisting of an overview of the relevant background material and a state-of-the-art review of recently developed systems in this field. The methodology applied in this study will then be described, and the performance of the proposed systems will be then be outlined. Following this, the result will be analysed in-depth, and the limitations and impact of this research will be identified and explored, before addressing areas for improvement in future work. Finally, the project will be summarised with concluding remarks.

CHAPTER

TWO

LITERATURE REVIEW

2.1 Introduction

This literature review presents an overview of the relevant background material, along with a comprehensive review of current state-of-the-art publications in spine and vertebrae segmentation models.

2.2 Background

2.2.1 Zebrafish as Animal Models

Animals are essential resources for addressing fundamental scientific questions and have contributed to several advancements in the development of new therapeutic solutions for skeletal diseases in humans [1]. Previously, mammals with close evolutionary relationships to humans were used for research purposes including mice and non-human primates. Their use as models has reduced over time due to a number of limitations surrounding associated handling

costs and ethical concerns with regards to research studies performed at early developmental stages. Most countries employ the mandatory "Three R's" principle whereby animals used in research must be replaceable, reduced and refined [2]. These limitations led to the increased use of alternative models in the study of human diseases, including zebrafish.

Zebrafish show several advantages over mice, their embryos are externally fertilised, transparent and exhibit rapid development [1, 3, 4]. The optical clarity, as highlighted in Figure 2.1, and ease of access to the early embryo facilitates genetic manipulation and visualisation of cell growth stages in vivo. The statistical power of experiments involving zebrafish is extremely high due to the large volume of offspring a single pair of zebrafish can produce (up to 300 embryos each week) [5]. Zebrafish and humans share 70% of all genes and up to 85% of diseased genes [4, 5]. This makes it possible to study disease related genes using zebrafish. The combination of these key characteristics, their small size and low maintenance cost have allowed zebrafish to become a popular choice as an alternative model for the in vivo study of human conditions since the 1970's [6].

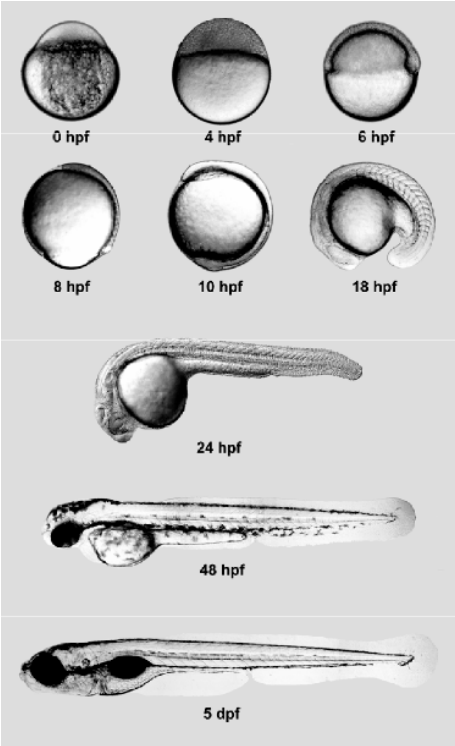


Figure 2.1: Zebrafish embryonic and early larval development over subsequent hours post fertilisation (hpf), adapted from [7]

Zebrafish are genetically traceable models that have been used to investigate ranges of human diseases, from skeletal conditions to addictions, as a direct result of the high synteny between the genomes of zebrafish and humans [4]. During early development, large scale mutagenesis experiments have been carried out on zebrafish to isolate mutant genes associated with skeletal disease defects in humans alongside high-throughput screenings identifying chemical compounds which could potentially revert pathological phenotypes [3]. This process generates hard tissue dysfunctions similar to humans and have previously been used to model osteogenesis imperfecta, scoliosis and osteoarthritis. The turnover and remodelling of these tissues can be studied in the long-term as mutated zebrafish grow, further developing pathological studies of adult human bone diseases and facilitating the identification of novel therapeutic targets. In the past, many human mutations have been expressed in adult zebrafish that have allowed for further studies into long-bone fractures, osteoporosis and even obesity [4].

Skeletal phenotyping of the mutated adult zebrafish follows similar traditional techniques, despite their small size [1]. In humans, X-ray imaging is one of the more frequently used techniques to visualise the skeleton. This process is repeatable on live organisms and is relatively cheap and quick. Classic X-ray systems limit their radiation exposure for the patient, with respect to both animals and humans, which subsequently limits the exposure of the image. As a consequence, regular medical appliances are not appropriate to obtain an adequate X-ray of the zebrafish. Within the last decade, X-ray machines have been adapted for smaller animals, allowing the acquisition of zebrafish skeleton images with higher resolutions. Visualisation of the zebrafish's skeletal system can also be performed using MicroCT, but it is expensive, time-consuming and generates large 3D data that is not easily analysed. Histological approaches are also frequently used to complement other imaging techniques or for cellular evaluation.

2.2.2 Introduction of Artificial Intelligence

AI is a subspecialty of computer science consisting of algorithms that deliver task-solving capabilities comparable to that of humans [8]. Machine Learning (ML), both automatic or autonomous, is a dominant subfield within AI which imparts the capacity of a computer to learn without explicit programming to make accurate predictions based on new data [9]. Recent developments in AI and ML have allowed for automatic problem-solving without human intervention. Improvements in computer vision technologies have allowed enhancements in the abilities of computers to detect and diagnose diseases present within local regions of medical images through the evolution of DL based on the concepts of neural networks. AI has exceeded previous expectations whereby it has shown capability in detecting, characterising and measuring complex pathologies in a variety of physiological systems, for example the musculoskeletal system. Where their uses are extremely convenient, diagnoses of medical conditions require careful analysis of local regions in one image [10]. ML algorithms and AI can further improve with more data and experience but with regards to the medical domain, it will always need guidance from its experts to ensure their reliability and validity [9]. From this, rapid advancements of AI and ML have been hindered in the field of medicine due to legal, ethical and moral aspects.

Deep Learning and Neural Networks

DL is a recent innovation in AI that has made huge advancements in the last decade alongside the improvements of computer vision [9]. Its use in image processing has further developed different visual tasks including classification, detection, and localisation [10] while also automating the analysis of various biological images [11]. DL has proven to be among the best models for such tasks, as their methods imitate the human recognition process that occurs in the visual cortex [12]. As a result, the technique is capable of recognising objects in natural world images and is highly effective at detecting abnormalities on medical images with sensitivities and specificities close to skilled clinicians [8]. In spite of the challenges met in medical fields,

DL approaches are quickly becoming methods of choice for analysis due to their rapid growth in other fields.

Convolutional Neural Networks (CNNs) are common approaches in the application of DL for image analysis [11]. The basic fundamental workings of a CNN is to take an input, perform mathematical operations on it over many layers and produce a desired output image. One CNN is characterised by a set of "learnable parameters" which provide the network with appropriate values (weights and biases) throughout its layers to allow it to adequately perform its given task. A basic schematic diagram of a CNN is shown in Figure 2.2.

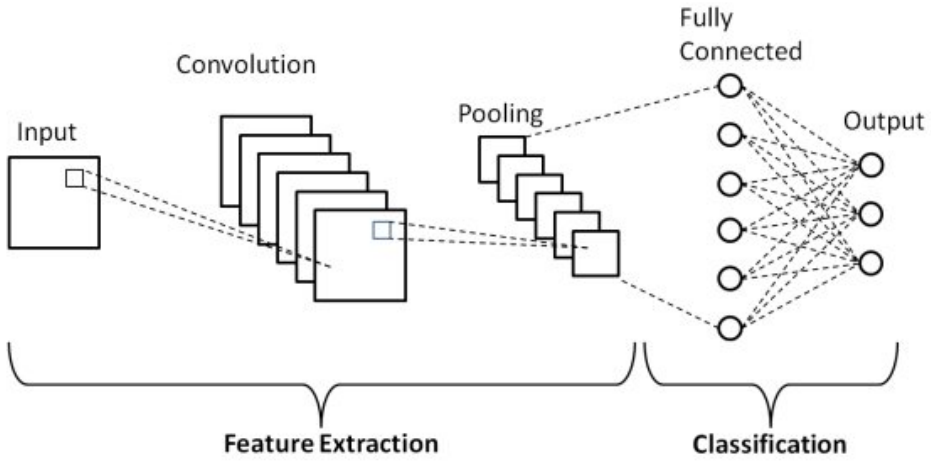


Figure 2.2: Schematic of a basic Convolutional Neural Network architecture, adapted from [13]

A CNN learns its characteristics through adequate training stages [11]. During this process, the CNN is given a set of input images and associated target output images, known as Ground Truths (GTs), while its learnable parameters are adjusted iteratively until the output of the network matches their targets appropriately. With respect to the medical field, GTs are generated through annotations made by medical experts that identify localised Region of Interests (ROIs) in the images. In order to develop a highly efficient and well trained model, the available GTs must be numerous and well-defined, especially for the purpose of aiding medical diagnoses [8, 9, 14]. CNNs are considered efficient and highly rated when they perform objectively and produce valid data consistently.

The success of CNNs were previously inhibited due to the limited size of available training sets and the associated size of the CNN (number of layers, parameters, etc.) [14]. Originally, training of CNNs relied on Central Processing Unit (CPU) based computations which were incapable of training deep networks, allowing other analytical techniques (for example, Support Vector Machines [15]) to become popular. These techniques could not meet clinician level performance and were deemed inappropriate [9]. The parallelism of a Graphics Processing Units (GPUs) could support the many inner-product operations of a CNN and convert them into a matrix operation [16]. Advancements in faster and cheaper GPUs allowed CNNs to rise in popularity as deeper networks with more layers could be trained. CNNs inevitably became popular computer models that were handcrafted to mimic human perception of different features of interest while availability of large labelled datasets also grew.

Limitations of Deep Learning Models and Neural Networks

As powerful as DL models are, they are not without shortcomings. Successfully trained DL models generated to detect and diagnose diseases require a large amount of varied training samples as well as appropriate training time and large computational power [14]. In the medical field, data generation is hindered due to ethical approval and because of the amount of valid data publicly available [9]. Robust models can be generated with scarce data through implementation of data augmentation (rotating and flipping samples) and transformations (adding different types of blurring or adjusting image contrast). Use of different architectures, such as the U-Net discussed in section 2.2.3 or training using GPUs, could also overcome this issue [10]. Adapting the CNN's structure by reducing the number of learnable parameters or ignoring parts of the network during training will reduce the time and computational power necessary for appropriate training.

GTs used for successful training of DL models must be annotated by medical experts to ensure they are well-defined [9]. This task is time-consuming and labour intensive for one person to annotate a large enough sample for appropriate training [8]. Inherent biases that

one human expert could have will heavily bias the output of the model [11]. Automating the annotation process could reduce the burden of having human experts handling large datasets and eliminate potential biases, but the GTs will still need to be validated. Using multiple annotators reduces the workload and time needed for data generation. Individual biases will subsequently be eliminated during training when the CNN focuses on the common features present across all GTs. Ensembling is another technique that can be applied across a series of models as visualised in Figure 2.3. This technique is particularly useful when all networks are trained on different datasets and can subsequently boost the final model's performances [13].

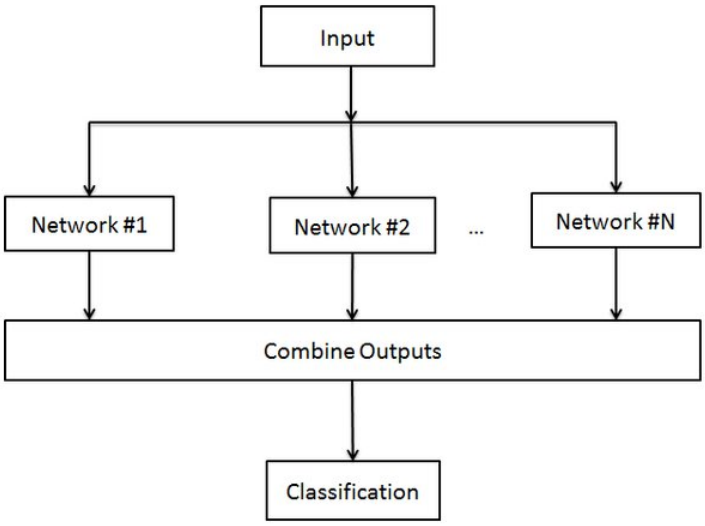


Figure 2.3: Ensemble of neural networks. Input is applied to all neural networks and their associated outputs are combined to generate one overall output, adapted from [13]

DL models appear as "black boxes" where the behaviour of internal hidden layers and computations are not visible, making them difficult to trust [10]. This causes problems for human experts and algorithm developers alike with regards to interpretation and replication of results [8]. Current ML algorithms are expected to train on the basis of a medical expert's annotations and alongside their interpolations. This issue has given rise to Explainable Artificial Intelligence (XAI), a subsection of ML which produces more explainable models with similar prediction performances in an effort to induce appropriate understanding and trust between medical experts and ML [17]. Gradient-weighted Class Activation Mapping (Grad-CAM), is an example of XAI which employs the use of the gradients from target images

connected to the final convolutional layer in a CNN and produces a coarse localisation map of import features in the original image that the network uses for its prediction [18]. This can be seen in Figure 2.4. Computer-aided diagnoses have mixed reputations with patient care and there is a reduction in enthusiasm for these techniques in clinical settings [9]. Despite the progress of DL, it has yet to be used in real-world medical trials until it can perform with minimal error as required for medical analysis [10].

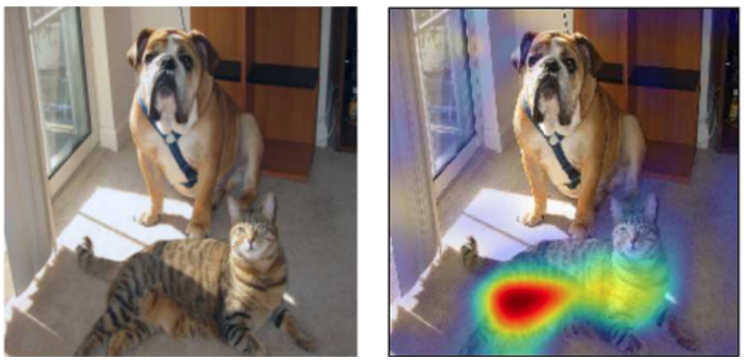


Figure 2.4: Sample localisation mapping ability of Grad-CAM. When an image of a cat and dog are inputted, the Grad-CAM network colours the features associated with the class label in the image. In this case, the class label was cat. Adapted from [19]

2.2.3 Applications of Artificial Intelligence

Image Segmentation

Image segmentation is a necessary first step for automated information extraction methods as it outlines and separates objects in images [10, 12]. Under this process, images are fragmented into a series of regions based on similar representative attributes which are further used to classify objects employing a variety of techniques, including ML [20]. Segmentation demonstrates great potential in medical imaging applications, as it distinguishes objects of interest from images, subsequently capable of extracting useful information from different anatomical structures. The task can be accomplished using several different techniques.

Previous Techniques

Classical methods include thresholding, region-based and edge-based methods [20]. These methods are relatively simplistic, where images are separated as a result of varying pixel intensities, whether it be due to them exceeding specific tolerances or exhibiting similar intensities.

Pattern recognition-based methods include supervised and unsupervised methods such as classification and clustering respectively [20]. Clustering techniques, such as the K-means algorithm, are unsupervised and similar pixels are grouped together with very little user-interaction. Classification techniques, such as k-nearest-neighbour classifiers, must be supervised as they require a training phase and are seriously affected by noise and poor data. Using these techniques, image segmentation in medical image analysis can reduce down to pixel classification problems with two labels, $\{$of interest, not of interest$\}$, but cannot provide pixel-level context information. As a result, they are rarely used in this domain [10].

Previous techniques are limited as a result of their structure and necessity for expert interaction [20]. This has led to the rise in interest for more knowledge-based methods/algorithms including the use of deformable models and CNNs. These techniques train models using domain specific rules to yield precise segmentation results. Deformable models perform segmentation based on the idea of moving curves or shapes as a result of external and internal factors. This includes Active Shape Models (ASMs) where models locate shapes in test images after rigorous training stages performed with other images. CNNs are typically used in classification tasks to reduce images into single labels [10, 14]. The architecture extracts feature characteristics from the inputted images and generates feature maps at each layer within its network through convolutions and additions of weights and biases to produce the final labelled image.

Impact

Accurate segmentation tools are of high interest within the medical imaging community [10]. CNNs have previously been built to differentiate bone tissue area from medical images and classify each pixel between {bone, not bone} [12]. X-rays propose complex natures and present various results due to different equipment that effect their orientations, resolutions, and luminous intensities [20]. Radiographs can become indistinct or disconnected as a result of noise, artifacts and spacial aliasing influenced by these aforementioned factors. By subdividing X-ray images into various portions relating to bone tissue through segmentation, medical experts can analyse the structure and composition of bone to find fractures, malformations and pathologies. The use of CNNs with X-rays are still relatively limited. As X-rays have large dimensions, the training time for particularly efficient CNNs are long. Where CNNs are essentially pixel classifiers, the label for each pixel must originate from square areas. To tackle these issues, X-ray segmentations generally focus on certain areas of maximum interest within small square regions as discussed in section 2.3.

U-Net Architecture

The U-Net is a popular CNN architecture commonly used as a segmentation tool in medical image analysis which transforms an image into one single label [10]. The architecture has the ability to train using very few training images while still producing precise segmentation results, with regards to pixel-wise accuracy and similarity coefficients [14]. This is of great importance to the medical domain where accurately labelled data can be scarce. Records show that the architecture has won many ISBI challenges and previously beaten the state of the ray with considerable margins. The evident success of the U-Net has allowed it to be widely used amongst all major image modalities including X-rays, CT, MRI etc., for example in diagnosing rheumatoid arthritis or osteoporosis from X-rays of bones [10].

As a CNN, the U-Net consists of a large number of mathematical operations, as seen in Figure 2.5. The input is fed into the architecture and data is propagated across all paths

to generate the final segmentation result [14]. It is based on two symmetrical stages, the contracting path (left) and expanding path (right) as visualised in Figure 2.5, allowing for a symmetric architecture. The contracting path consists of two successive 3x3 unpadded convolutions, each followed by a Rectified Linear Unit (ReLU) and a max-pooling layer for downsampling [10]. This is repeated several times and alongside each downsampling step, the number of feature channels doubles [14]. The U-Net's novelty arises where each step in the expansive path consists of upsampling the feature map using a 2x2 "up-convolution", subsequently halving the number of feature channels [10]. Corresponding feature maps from each layer are cropped and concatenated, followed by two successive 3x3 convolutions and ReLU activation. Cropping prevents the loss of border pixels in every convolution. At the final stage, a 1x1 convolution is used to reduce the feature map to the required number of channels and produce the segmented image. In total the network has 23 convolutional layers.

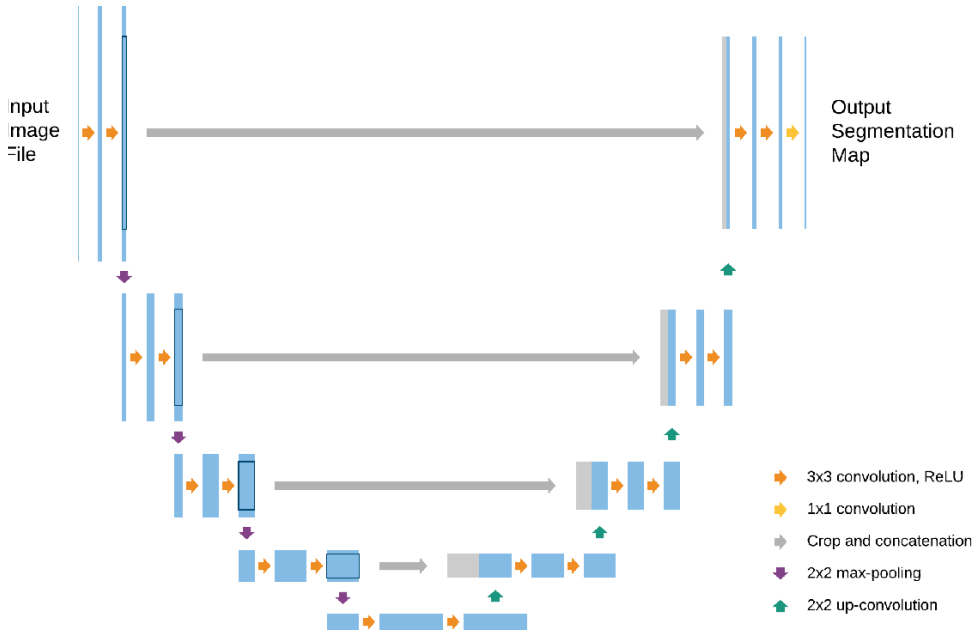


Figure 2.5: Basic U-Net Architecture. The arrows represent different operations, the blue boxes represent feature maps at each layer, and the grey boxes represent cropped feature maps from the contracting path, adapted from [10]

2.3 State-of-the-Art Review

Previous work involving CNNs and segmentation of spinal regions has focused on the spine in humans, with very little extending further to vertebrae and/or other animals. The results from these segmentations are further built upon to develop tools for automatic spinal landmark detection or quantification of spinal curvature in humans relating to the Cobb Angle. Overall, the field of segmentation and analysis of spinal regions in animals has not been well-researched, let alone that of zebrafish. This review reflects methods used previously to segment the spine or vertebrae using X-rays and subsequent methods of analysis applied.

2.3.1 Spine Models

The Cobb Angle is the gold standard measurement for spinal curvature diagnosis and treatment in humans [21]. Clinicians identify the Cobb Angle as the angle made between the most tilted vertebra endplates on X-rays, visualised in Figure 2.6 [22]. Measuring the Cobb Angle is deemed tedious and the accuracy of the angle measurement is dependent on the quality of the radiograph and the subjective experience of the radiologist [23]. These factors have allowed for maximum measurement errors of up to 10°, which are large enough to greatly influence the diagnosis of spinal curvature and associated treatments [24]. Publications have identified the need for a more robust framework that produces more reliable calculations resulting in many proposed segmentation methods to quantify spinal curvature in humans.

Wu *et al.* developed a Multi-View Correlation Network (MVC-Net) based on Anterior-Posterior (AP) and Lateral (LAT) view X-rays to generate multi-view spinal curvature estimations using a fully automated end-to-end framework [22]. The MVC-Net consisted of three parts: a series of cross-linked convolutional layers (X-Modules) for joint representations of spinal structures, a spinal landmark estimator network and a Cobb Angle estimator network for accurate angle estimation. The ensemble framework is depicted in Figure 2.7.

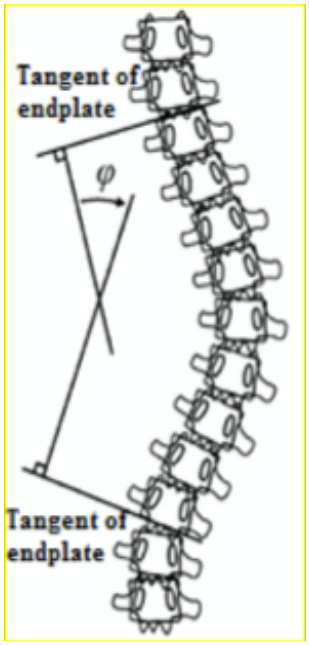


Figure 2.6: Cobb Angle Method: Estimation of the angle of spinal curvature, φ, which is defined by the two tangents of the upper and lower endplates of the upper and lower end vertebra respectively, adapted from [25]

526 AP and LAT spinal X-rays were collected and scaled to dimensions 128 x 256 pixels [22]. Each X-ray depicted 15 vertebrae from the thoracic and lumbar spine and the four corners of each vertebrae were identified by experts allowing for 60 landmark GTs per spine. Cervical vertebrae were ignored as they were too high to be involved with the spinal deformities investigated in this study. The dataset was augmented to enhance the performance of the model. Random Gaussian noise was introduced to stimulate inherent noise across the dataset. Images were rotated through 355° in intervals of 5° to make the framework flexible through rotations. Images were shifted by 1° to encourage shift invariance in the model. Networks were implemented in Keras.

The MVC-Net framework produced accurate and robust spinal curvature estimations for multi-view X-rays based on average performances over 10-fold patient-wise cross-validations [22]. The framework achieved a low average Mean Absolute Error (MAE) in identifying spinal landmarks (0.0398 AP and 0.0497 LAT) which further associates with a high Pearson Correlations (0.956 AP and 0.945 LAT). These results further conceded Circular MAE scores of 4.04° for AP and 4.07° for LAT Cobb Angles. These metrics are significantly smaller

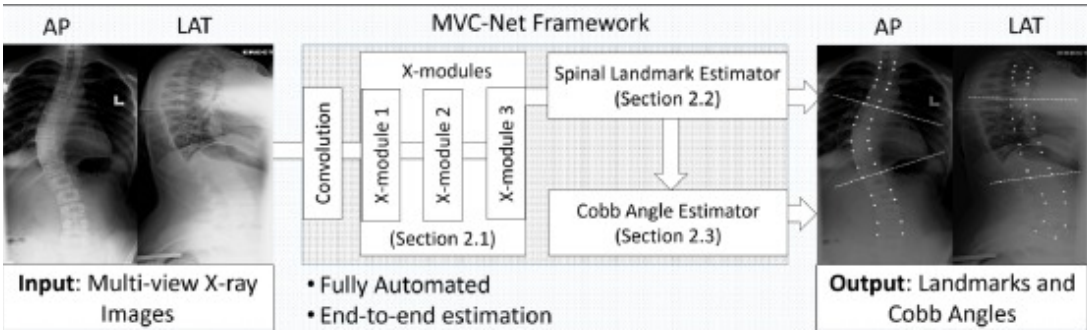


Figure 2.7: Proposed architecture of MVC-Net for Cobb Angle estimation using AP and LAT X-rays, adapted from [22]

than the error as reported with previous manual measurements. Where these results reflect the framework's potential as a helpful tool for clinicians in the evaluation of scoliosis, this study was limited and still should not replace the opinion of the domain expert. The dataset did not report variation in the diagnoses of spinal curvature or depict metal artifacts, such as spinal bracing, which could cause a reduction in spinal landmark accuracy and potentially result in a poor estimation of the Cobb Angle.

Wang *et al.* proposed the development of a Multi-View Extrapolation Net (MVE-Net) based on AP and LAT view X-rays allowing for multi-view automated scoliosis estimation, similar to the MVC-Net [21]. The MVE-Net consisted of three parts: a joint-view net which learned common landmarks between views, an independent-view net which learned from each view independently and the inter-error correction net which offset the previous network errors for accurate angle estimation. The automatic framework is depicted in Figure 2.8.

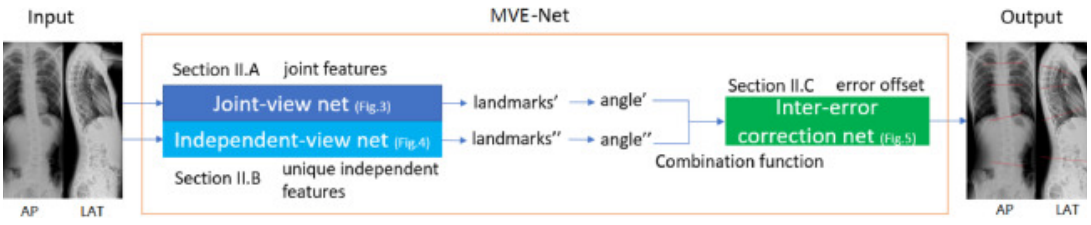


Figure 2.8: Proposed architecture of MVE-Net for Cobb Angle estimation using AP and LAT X-rays, adapted from [21]

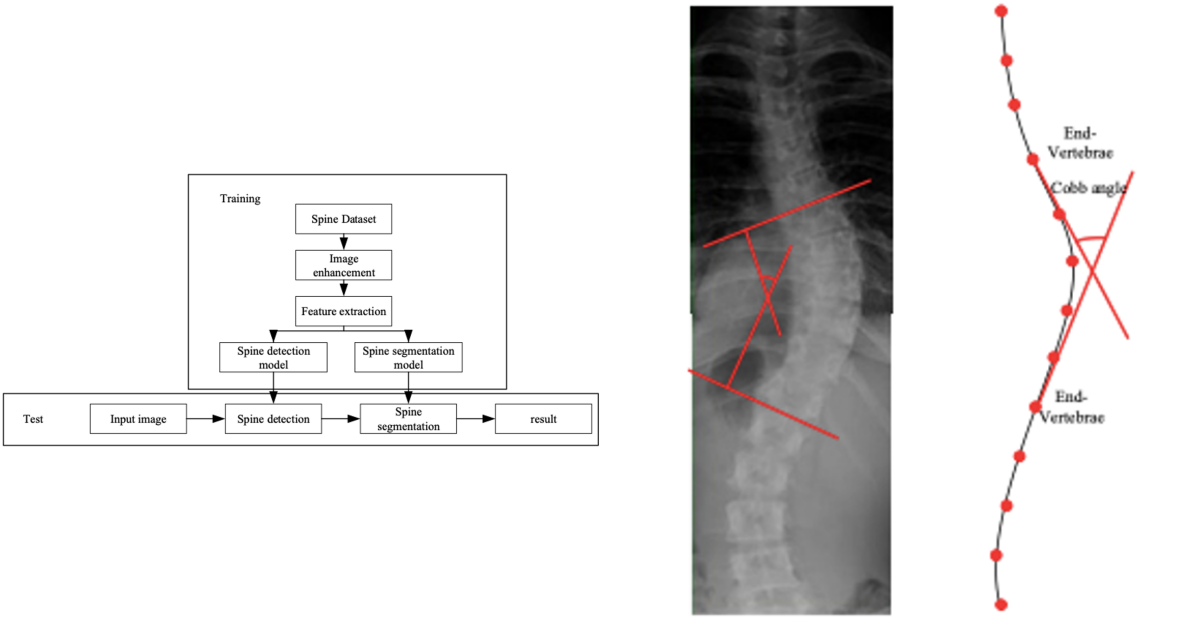
526 spinal X-rays of both AP and LAT view each depicting scoliosis and spinal deformities of varying extents were provided by local clinicians for the study [21]. Similar to the work of Wu *et al.*, each X-ray depicted 15 vertebrae from the thoracic and lumbar spine and four

corners of each vertebrae were identified by experts allowing for 60 landmark GTs per spine. Cobb Angle measurements varied between 0° and 96.33° across the dataset. The dataset was augmented to enhance the performance of the framework. Images were rotated through 355° in intervals of 5° and shifted by 1° such that the model was rotation and shift invariant, similar to the work of Wu *et al.* [22]. Random Poisson noise was implemented throughout to stimulate intensity variance. Networks were implemented in Keras using a Tensorflow backend and were trained on four NVIDIA Tesla GPUs with a version of CUDA 8.0.

The MVE-Net produced accurate spinal curvature estimations for multi-view X-rays based on an unseen test dataset [22]. The framework achieved a Circular MAE of 7.81° for AP and 6.26° for LAT Cobb Angles as well as a Symmetric MAE of 24.94% for AP and 11.9% for LAT. The variety of a large range of scoliosis measurements and implementation of an extrapolation layer minimised inter-error correction which allowed the framework to yield such high precision. As with Wu *et al.*, the work here can be used as a tool for domain experts but cannot be used as a replacement for diagnostic applications.

Tu *et al.* generated a Deformable U-Net (DU-Net) detection and segmentation network to segment the spine contour in a spine X-ray which automatically measured the Cobb Angle using spinal curve tangents [23]. The framework consisted of three subtasks: spine detection using aggregated channel features further trained using the Adaboost algorithm, spine segmentation using the DU-Net and spinal curvature quantification based on the tangent curves of a 6^{th} order polynomial fitted to the spine segmentation contour. This is visualised in Figure 2.9. 100 AP X-rays depicting scoliosis were obtained from the Anhui Medical University and manually segmented using "labelme". Cobb Angle measurements yielded from the framework were verified by orthopedic experts. The network was based on a PyTorch framework, implemented in Python and accelerated with a GPU.

The DU-Net segmentation framework yielded a precision of 0.86 and average Dice Similarity Coefficient (DSC) of 0.90. From this, the spine contour polynomials of the resulting segmentations fit the spines appropriately such that an MAE of 2.9° was calculated



(a) Segmentation algorithm flow chart **(b)** Cobb Angle estimation with polynomial fitting.

Figure 2.9: Framework of Deformable U-Net architecture and Cobb Angle estimation, adapted from [23]

when relating the tangent Cobb Angles to the manually measured labels. Despite this high performance, the use of the DU-Net is limited in a clinical setting as it deviates from current practice by not relying vertebral end-plates to generate its curvature estimations [22].

Publications relating to segmentation tasks with human X-rays are plentiful due to demand and the higher quality of resources in existence. With regards to animals, the availability of these solutions reduce as a result of the poorer image quality and further again with size [25]. Identification of perturbations in the skeletal structures of mice are significantly relevant to the studies of the Mouse Genetics Project where mammalian genes are systematically removed and screened over a range of resulting traits.

Oshaki *et al.* proposed an automatic solution for spine segmentation and curvature quantification of mice X-rays to automate the screening process, reducing processing time and costs due to human resources and error respectively [25]. The framework consisted of three parts: preprocessing of the X-ray images to identify the spinal ROI, spinal segementation through Otsu thresholding and other morphological processes, and spine curvature quantification determined by angles defined by extreme points of the spine contour and points of

intersection across a reference spine model. The framework is visualised in Figure 2.10. 100 dorsoventral mice X-rays were randomly selected from the Wellcome Trust Sanger Institute. Samples generated reflected different variations of scoliosis (none, thoracic, thoracolumbar, lumbar) as defined by domain experts. The effectiveness of the resulting spinal curvature measurements were then compared to manual measurements performed by experts and 6 other automatic curvature methods. From this work, a public toolbox was developed in MATLAB that employed the automatic techniques outlined to produce 6 possible estimations for spinal curvature in mice with the ability to distinguish abnormal shaped spines with an accuracy of 98.6%.

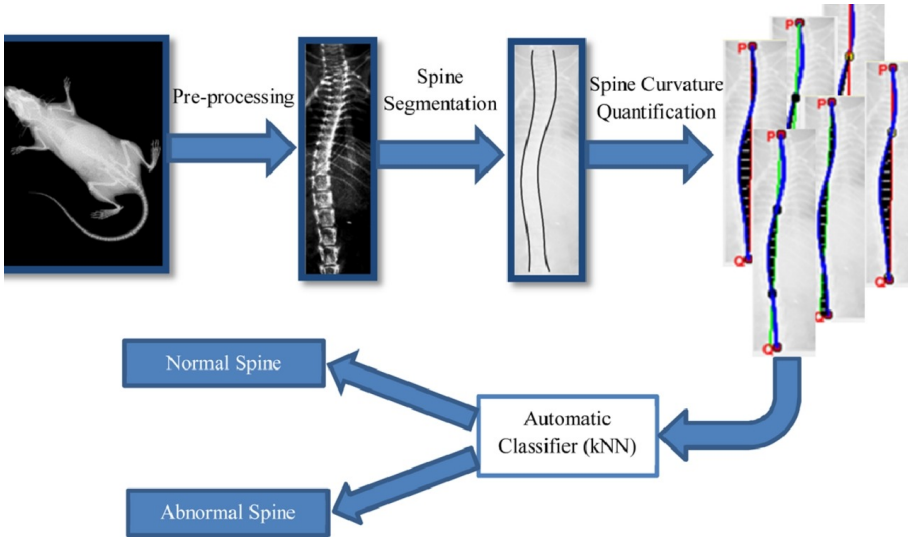


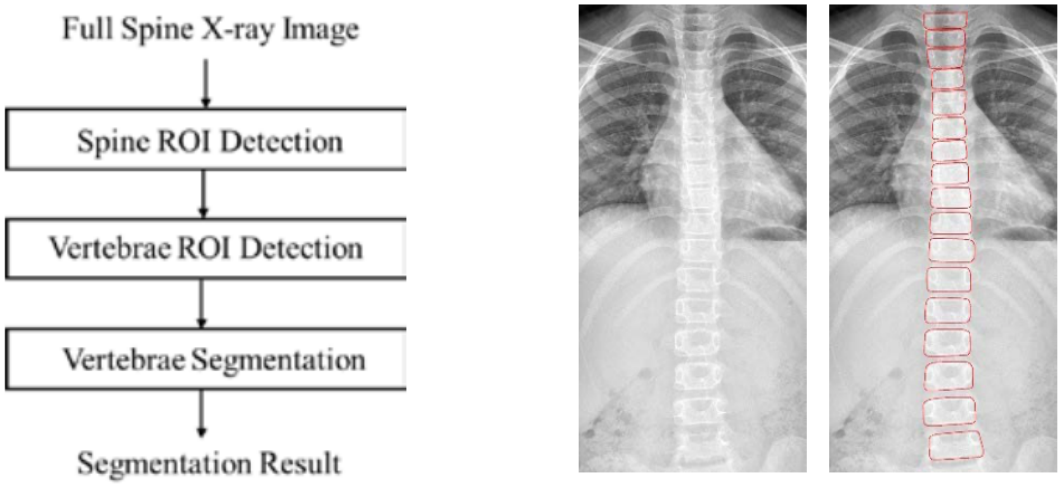
Figure 2.10: Automatic mice spine curvature estimation framework, adapted from [25].

2.3.2 Vertebra Models

The spine is an important structure in the body and injury/critical deformation to it can cause severe pain and immobility [26]. Approximately 50% of spinal cord injuries happen in the cervical region as a result of its high range of motion and flexibility and 20% of these injuries can go unnoticed during radiological examinations as a result of human error [27]. Over time, these can deteriorate even further to become life-threatening. Accurate segmentation of vertebrae in X-rays can be used by clinicians to quickly analyse the severity of diseases

and injuries in the spine. Previously, this had been accomplished using statistical shape model based approaches [28] as outlined in [29] and [30]. DL methods previously had varied results for this task, but the following works have produced meaningful results by analysing space patches where vertebrae are and produce a segmentation mask for that region [26, 27, 28]. In both cases, the vertebrae shape was preserved using novel shape-aware loss functions.

Kuok *et al.* developed a framework that successfully segmented the thoracic and lumbar vertebrae of AP X-rays [26]. The hybrid framework consisted of three steps: identification of the spinal region using Otsu thresholding, identification of individual vertebral regions through analysis of pixel intensities over a polynomial fitted to the central line of the spine and the segmentation of individual vertebra using the U-Net architecture. The methods are visualised in Figure 2.11.



(a) Segmentation algorithm flow chart **(b)** Resulting boundaries of vertebra segmentations

Figure 2.11: Proposed thoracic and lumbar vertebrae segmentation framework, adapted from [26].

60 AP spine X-rays of young scoliosis patients were obtained and digitised with varying gray scale pixels [26]. GTs of thoracic and lumbar vertebrae were annotated by domain experts. The U-Net designed specifically for vertebra segmentation was implemented in Python using a Tensorflow framework, and trained using a PC equipped with a NVIDIA GTX1080Ti display

card and Intel i7 CPU. The framework's performance was evaluated using a five-fold cross-validation on the ROI vertebra patches previously obtained [26]. For such a complicated segmentation task, it yielded an average DSC of 0.941 while being trained on a small dataset showing the promise of the proposed method.

Al Arif *et al.* outlined a fully automated framework for the segmentation of cervical vertebrae in human X-rays [27]. The proposed method consists of the concatenation of three CNNs, each focused on one task. The first CNN localised the spinal cervical region as an ROI. The second CNN predicted a probability distribution map which identified the centers of vertebrae C3 to C7. The resulting outputs were merged together and inputted into the final shape-aware CNN which segmented the vertebrae. The proposed framework is visualised in Figure 2.12.

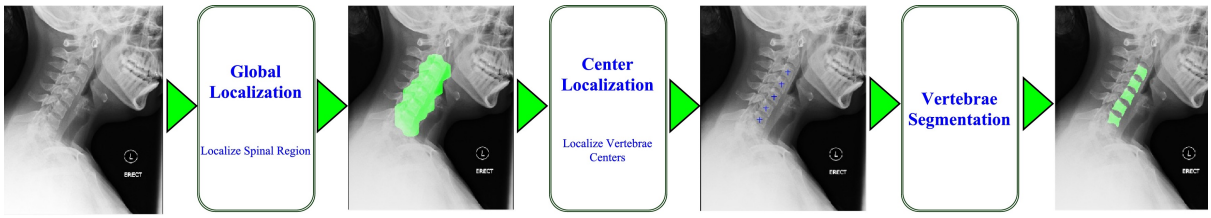


Figure 2.12: Fully automatic cervical vertebrae segmentation framework, adapted from [27].

296 cervical region X-rays of various sizes and resolutions were collected from real-life hospital emergency rooms [28]. GTs of respective cervical regions, vertebrae centers and vertebrae shapes for each image were manually annotated by domain experts [27]. Of this dataset, 124 X-rays were used to train each subsection of the framework. This was augmented through a series of rotations to make the final model rotation invariant. Networks were efficiently trained on NVIDIA GPUs.

The performance of the framework was tested on 172 unseen X-rays and yielded an average DSC of 0.84 and vertebra shape error of 1.69 mm [27]. The final CNN was trained using individual patches of X-ray depicting a single vertebra. Alone, the shape-aware segmentation model yielded an average DSC of 0.944 and an average point to curve error of 0.55 mm, outperforming previous ASM-based methods [28]. As the output of the automatic

framework relied on the X-ray patch generated by the centre localisation task, the final model's performance was poor in comparison to the final task alone. This result could have been improved by incorporating a step which removed outlier centers away from the vertebral curve.

2.3.3 State-of-the-Art Conclusion

As reflected in this review, research publications relating to spinal region segmentation and subsequent analysis are heavily dominated by studies for humans alone. However, the work reviewed here shows strong potential for use of segmentation tools and neural networks in their respective domains.

The study of automatic spine segmentation models using DL methods in humans is limited due to severe lack of available data and reliability for analysis in the medical field. Taking the Cobb Angle as an example, this metric for spinal curvature quantification specifically relates to the orientations and positions of the vertebrae. By segmenting the spine, the information of the vertebrae is lost and renders the model useless for clinical applications and diagnoses as seen in [23]. From this, automatic methods of spinal landmark detection are more popular even if their performances are not as strong because they are based on clinical definitions, as seen in [22] and [21]. However, with genetic screening of animals, the phenotype or effect of a gene mutation is of more interest than the clinical "human" diagnosis or exact measure. This allows spine segmentation models using DL and animals to be a powerful tool for the acceleration of the genetic screening process as seen in [25].

Automatic vertebra segmentation models are of huge benefit to the medical community as they can outline the shapes of vertebrae objectively to quickly analyse the severity of diseases or injuries to the spine. However, this greatly relies on the model's ability to preserve the specific shape of the vertebra through its processes. The performances of the models reviewed here greatly rely on the ROI proposed in previous steps. In this case, the apparently simpler ROI based approach outlined by Kuok *et al.* outperformed that of Al Arif *et al.*'s probability distribution as it relied on the positioning of the spine reference line alone rather than several

similar landmarks in the cervical spine region [26, 27, 28]. There was a lack of literature relating to automatic vertebra segmentation models with animals, due to the lack of potential resources, high quality data and general interest. This is a disadvantage, with regards to the screening of genetic loci relating to skeletal condition as it has huge potential if successfully implemented.

Across the literature reviewed here, sufficient results were yielded despite varying data types, sample sizes, image preprocessing techniques and neural network implementations. Frameworks reflected strong performances when small datasets were well defined and limited to a specific region [23, 26] but larger datasets had the ability to incorporate more variety in the training process to make their frameworks more robust [22, 21]. Smaller datasets could be enhanced using a variety of augmentation techniques as highlighted in [27] and [28]. Despite the different applications and results of the frameworks reviewed here, they all relied on adequate and efficient training supported by GPUs.

2.4 Conclusion

The use of AI in medical fields has advanced in recent years but has been limited to being a tool for clinicians in the aid of diagnoses due to legal and ethical issues. AI has the potential to be used with animals but has been limited despite large datasets being easily accessible as it is a niche field. It requires a lot more collaboration as many experts from different fields must be involved in its discussion. Processes for reliable genetic screening of therapeutic targets can accelerate by merging DL techniques previously developed for spine and vertebra analysis in humans with the large and readily available dataset of mutated zebrafish. This study strives to develop a system capable of generating powerful metrics to analyse the resulting phenotypes of genetically modified zebrafish. More work on this study can further generate a framework to compare bone mineral densities between samples to test osteoporosis.

CHAPTER

THREE

METHODOLOGY

3.1 Introduction

This chapter outlines the development of each individual section of the novel pipeline generated for the study of the zebrafish skeleton. The implemented framework is outlined visually in Figure 3.1. Each section and design implementation was developed in collaboration with Dr. Erika Kague. Every segmentation mask and automatic measurement produced was verified and validated by the expert geneticist. Attached in chapter A are reference diagrams of the skeleton and anatomical planes of the zebrafish.

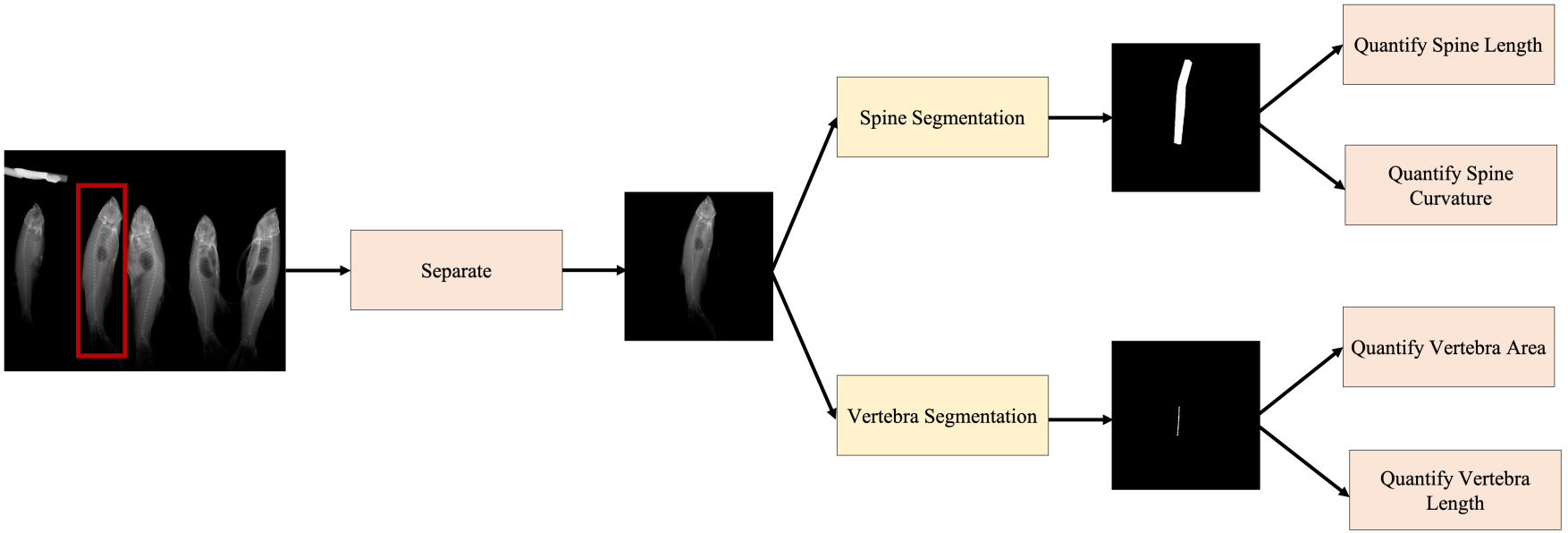


Figure 3.1: Summary of implemented framework generated through this study to rapidly analyse and quantify different phenotypes in zebrafish due to genetic manipulation. Orange boxes identify semi-automatic/automatic frameworks developed for analysis and yellow boxes identify neural networks generated for semantic segmentation.

3.2 Image Pre-processing

Live adult zebrafish were anaesthetised with MS222 prior to imaging [31]. Multiple samples were lined up vertically and radiographed using a MultiFocus digital radiography system (Faxitron). The same settings were used for each X-ray: 45kv, 5 seconds of exposure and 0.46 mA [32]. X-rays were handled as DICOM files with sizes of 4800 pixels by 6080 pixels. Calcium Hydroxyapatite phantoms (CaHAs) of densities 0.25 and 0.75 g/cm^3 were added in the top left corner of each X-ray for grey value calibrations. A framework was developed in MATLAB to automatically remove the CaHA and generate X-ray samples of individual zebrafish based on the predefined location of the CaHA sample. The steps governing the processes of the semi-automatic framework are outlined in Figure 3.2. The steps are further visualised in Figure 3.3.

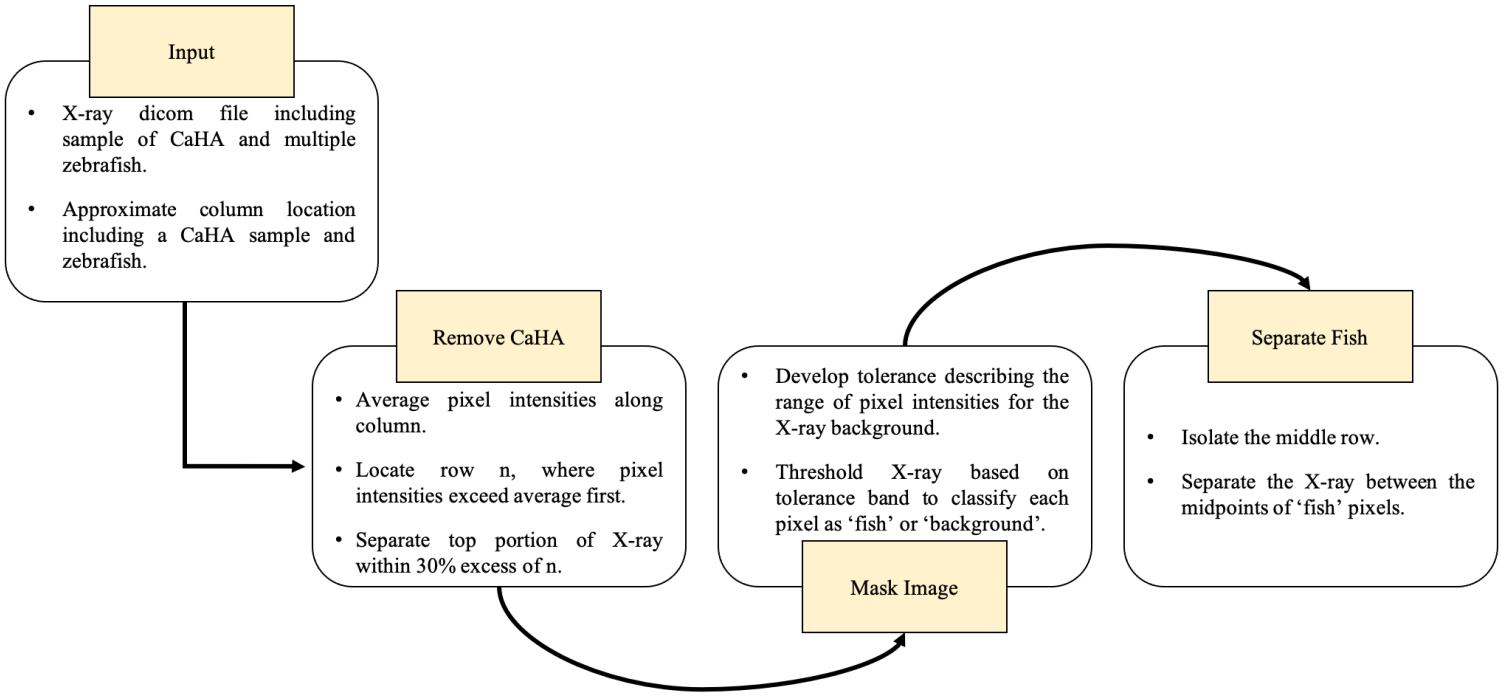


Figure 3.2: Summary of the processes carried out by the semi-automatic framework developed to remove the Calcium Hydroxyapatite phantom sample from X-rays and further generate X-ray samples of one individual zebrafish.

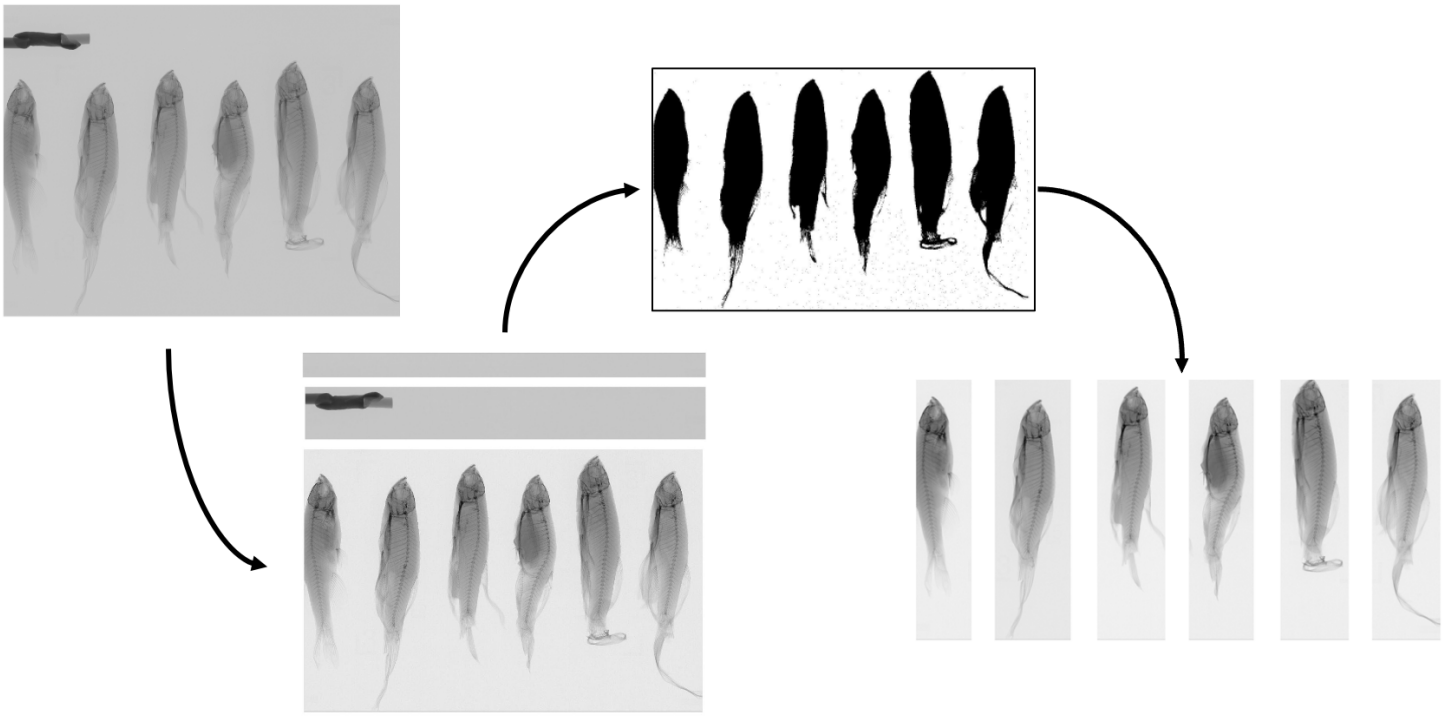


Figure 3.3: Visualisation of the processes carried out by the semi-automatic framework developed to remove the Calcium Hydroxyapatite phantom sample from X-rays and further generate X-ray samples of one individual zebrafish.

The first zebrafish on the left, shown in Figure 3.3, was the same fish in each X-ray. This zebrafish was known to be 28 mm. It was determined that the average pixel length from mouth to the end of the caudal fin vertebrae - when analysing the masked images generated from separating the original DICOM X-rays - for this fish was 2753.5 pixels, as outlined in Figure 3.2. This allowed for the following length ratio:

$$0.01017mm = 1pixel \tag{3.1}$$

A dataset of 812 X-rays of individual zebrafish was curated. The dataset varied in terms of the size of X-ray samples, orientation of zebrafish and phenotypes as a result of genetic manipulation. 612 of these X-ray samples were reserved for training purposes, while 100 samples were used for validation and testing each. Within the total dataset, 130 samples were identified as “difficult” cases due to overcrowding of samples in a single X-ray causing

poor separation results or general skeletal phenotypes (vertebral fusions, spine curvature, intervertebral disc diseases, etc.). These have been shown in Figure 3.4. 110 of these "difficult" cases were randomly selected and reserved for training purposes, while the remaining 20 cases were split randomly between validation and testing datasets.

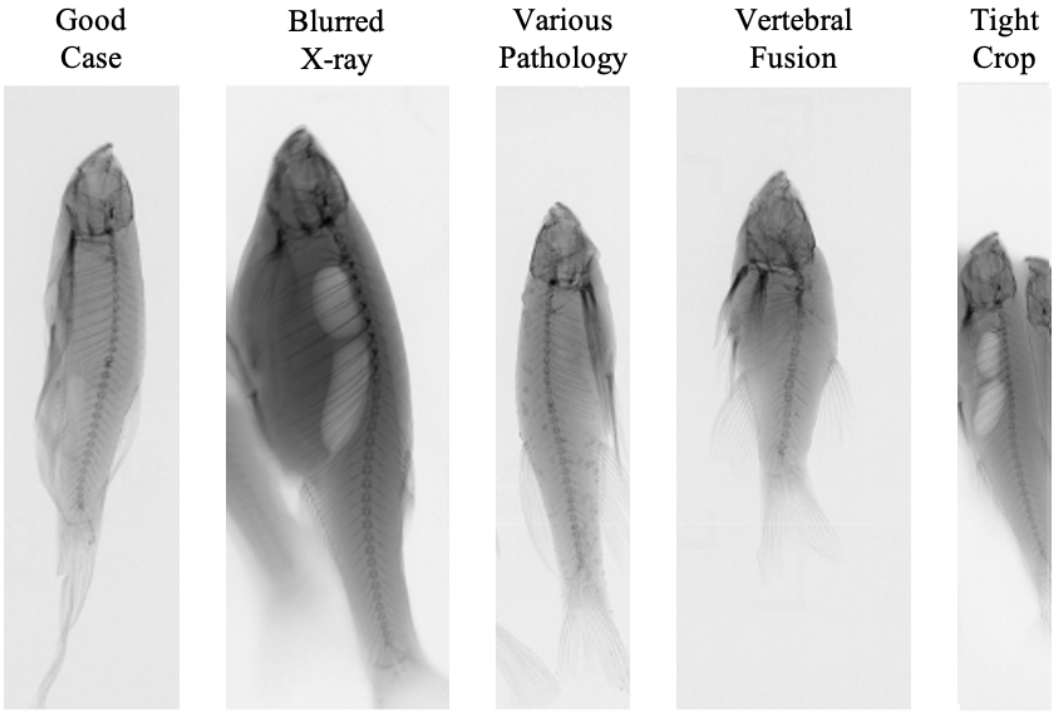


Figure 3.4: Visualisation of the variety of X-ray samples available in dataset. A "Good Case" was defined as a sample that included one zebrafish present in the centre of the X-ray. "Difficult" cases arose as a result of poor X-ray quality/movement, various pathologies (spots and vertebral fusions) and close proximity of samples during image taking.

3.3 Automatic Segmentation Networks

3.3.1 Data Generation

Spinal region Ground Truth

The spinal region was defined from the most posterior part of the skull connecting to the vertebral column, to the most caudal extremity of the zebrafish spine, proximal to the caudal fin. This is visualised in Figure 3.5. Refer to Figure A.2 for a labelled diagram of the axial skeleton of a zebrafish.

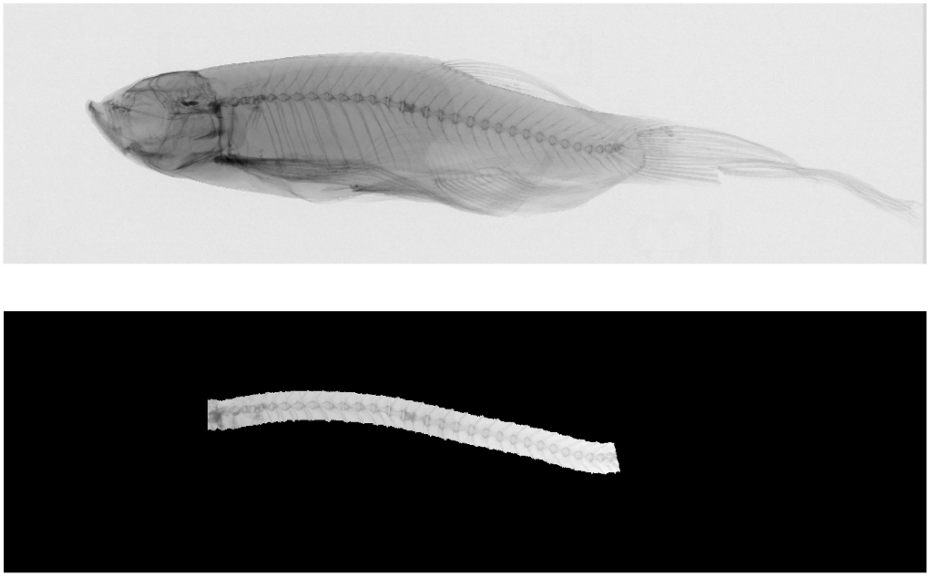


Figure 3.5: Visualisation of spinal region of interest on zebrafish. Spine segmentation mask encompasses the all vertebrae from most posterior part of the skull connected to the vertebral column, to the last caudal vertebra of the zebrafish spine.

GTs for spine segmentations were generated semi-automatically using a framework previously developed for basic image analysis on the zebrafish spine, developed by a collaborator of Dr. Kague - Yushi Yang [33]. The process has been outlined in Figure 3.6. "Difficult" cases that could not be correctly segmentented using this framework were manually segmented using the Image Segmenter App on MATLAB. On average, each segmentation of the spine GTs took 7 minutes. All GTs were validated by the expert geneticist.

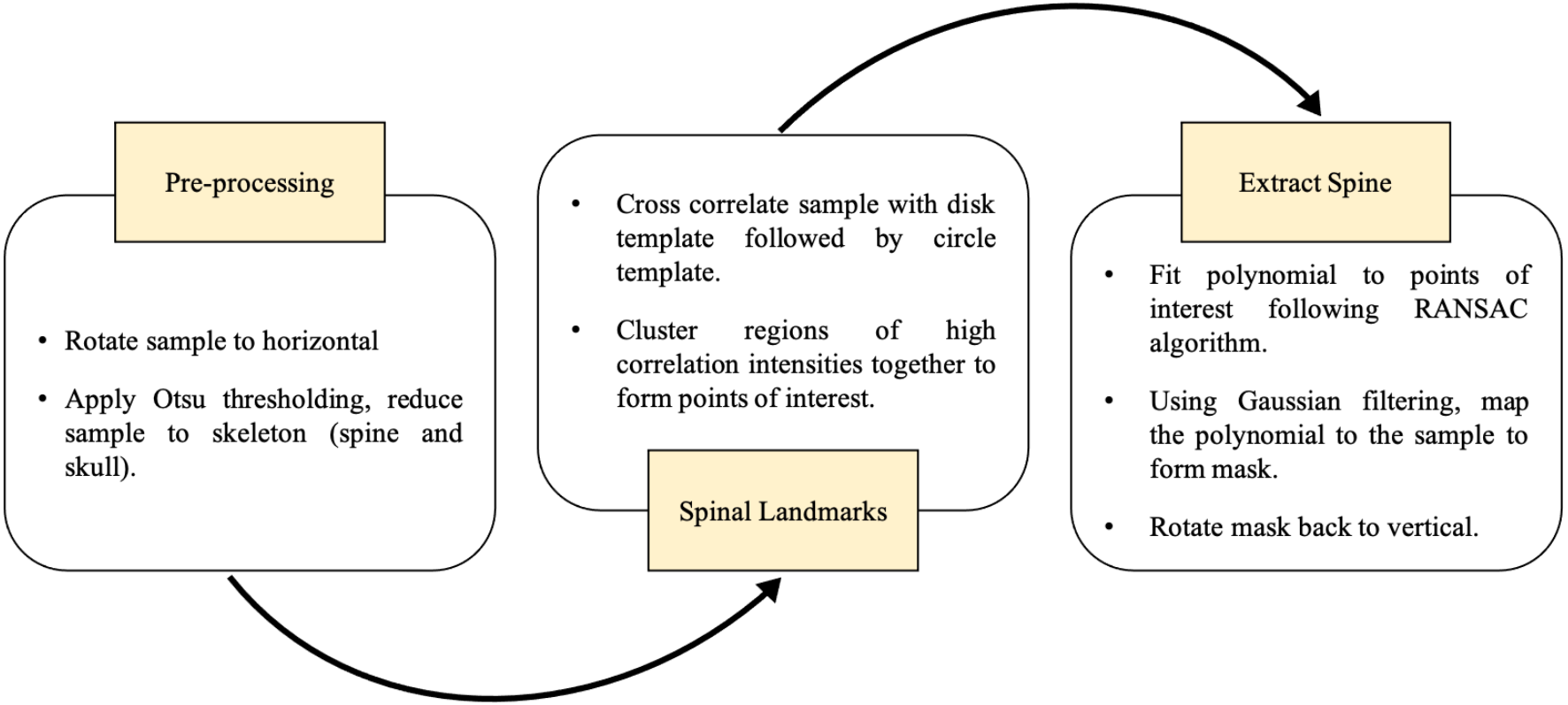


Figure 3.6: Summary of processes employed by the semi-automatic framework to generate the spinal region segmentation ground truths.

Vertebrae Ground Truths

To maximise the performance of the vertebrae segmentation model, the ROI was confined to only the space below the ribs and between the dorsal and anal fins. Refer to Figure A.2 for a labelled diagram of the axial skeleton of a zebrafish. One vertebra was defined as one bone segment between two intervertebral discs. For each X-ray sample, 10 vertebrae segments were manually annotated as GTs by 5 experts using the Image Segmenter App on MATLAB. Samples are shown in Figure 3.7. "Difficult" cases were referred to and annotated directly by the expert geneticist. On average, it took 10 minutes to manually segment a GT sample with 10 vertebrae. All GTs were validated by the expert geneticist.

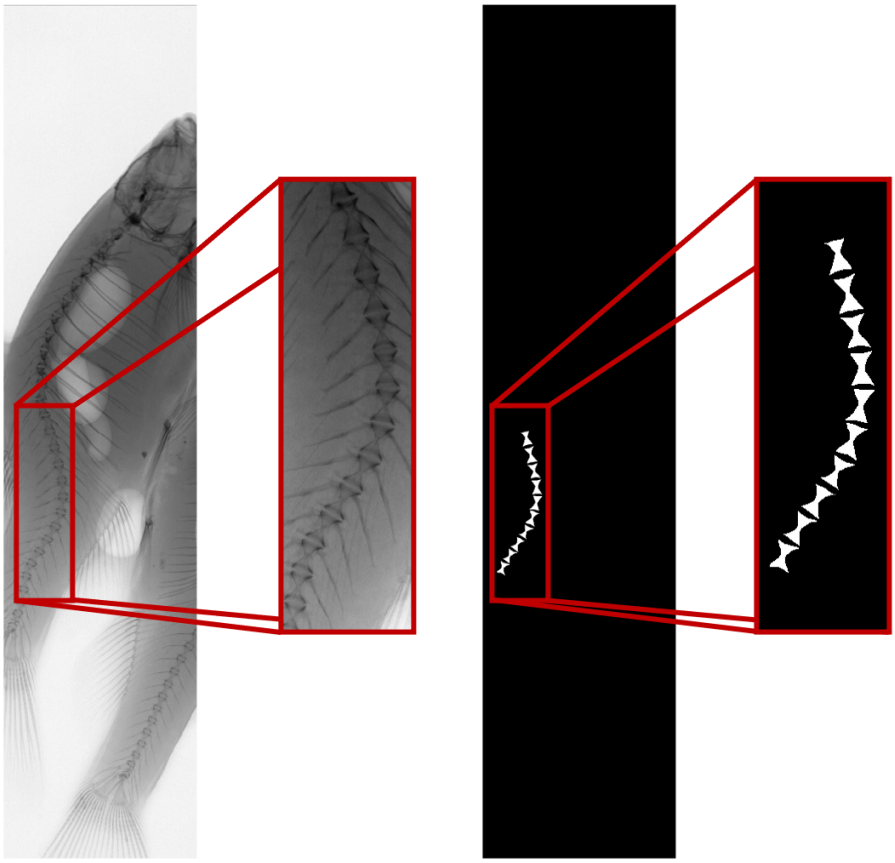


Figure 3.7: Visualisation of the 10 vertebrae segmentation ground truths identified for each X-ray sample. Vertebrae of interest were identified by first locating the vertebra connected to the final rib and manually annotating the next 10 following vertebrae, as per the expert geneticist's instructions.

Data Augmentation

In order to further enhance the training dataset and allow the model to generalise well to unseen data, the training dataset was augmented 26-fold using the following techniques:

- Each sample was rotated between $\pm$ 30° with 5° intervals. This increased the dataset 13-fold and allowed the model to be flexible with regards to the rotations it will be subject to.
- Each sample was flipped horizontally. This increased the dataset 2-fold and allowed the model to be flexible to different orientations.
- Varying ranges of Gaussian noise were distributed across samples. This introduced random inherent noise across the dataset with a kernel size of 3x3.
- Varying ranges of gamma correction were applied across the dataset to randomly alter the pixel intensities.

3.3.2 Neural Network Architecture

The U-Net discussed in section 2.2.3 was applied for semantic segmentation of the spinal regions. This reduced the segmentation problem to pixel-level classification, identifying each pixel as part of the spinal region or background. The U-Net architecture of [34] was adapted for this study. The applied network consisted of four down-sample blocks in the encoder section which were symmetrically built up in the decoder section. The layers were concatenated along the channel dimension to merge them together. More details of each layer are provided in chapter B. In total, there were 1,928,289 parameters to train.

3.3.3 U-Net Training

Network inputs and target outputs were zero-padded into a square area before being resized into tensors of size [256, 256], to allow for efficiency in training time and computational operations. All models were implemented in Python using a PyTorch backend and trained exclusively on the NVIDIA Tesla K80 GPU available in Google Colaboratory Pro [35, 36]. Due to restrictions on the Google Colaboratory Pro server, models could only train on the GPU for a maximum of 24 hours. Training was optimised using the Adam optimizer due to its history of being computationally efficient, allowing it to work optimally with larger datsets - as used in this study - while requiring little memory space [37]. The loss criterion "BCEwithLogitsLoss" was used to calculate the training and validation loss between respective datasets after each epoch [38]. It was appropriate for the datasets used in this study as the network inputs were arrays with a large range of values between [0, 65,535] and the targets of the network output were binary outputs. This loss criterion calculated the losses using a binary cross entropy loss function combined with a sigmoid layer. This approach was numerically more stable than using a plain Sigmoid followed by a "BCELoss" criterion as all operations were combined in one layer. Both the optimizer and loss criterion were available within PyTorch. Each model was validated using the same validation dataset from section 3.2 after every epoch to prevent over-fitting of the model.

Spine Segmentation Model

The final spine segmentation model was an ensemble based on the majority vote of the average output of four different segmentation models. Each model was trained using the same training dataset available in section 3.2, but under different hyperparameters. This includes different training sample sizes from the available training dataset and different batch sizes. The training hyperparameters of each model are shown in Table 3.1.

Model Number	1	2	3	4
Sample Size	306	306	612	612
Batch Size	10	16	16	16
Learning Rate	0.005	0.005	0.1	0.1
Epoch Number	25	11	10	12
Gaussian Blur Range	[0.1 - 2]	[0.1 - 2]	[0.1 - 1]	[0.1 - 1]
Gamma Contrast Constants	[0.25, 0.5, 1, 1.5, 2]	[0.25, 0.5, 1, 1.5, 2]	[0.25, 0.5, 1, 1.5]	[0.25, 0.5, 1, 1.5]
Training Time (hours)	19	15.5	15	18

Table 3.1: Training hyperparamaters of constituent spine segmentation models. A variety of hyperparameters were imposed to produce differently trained models for the final ensemble model.

Vertebrae Segmentation Model

The final vertebra segmentation model was also an ensemble based on the majority vote of the average output of four different segmentation models. Each model was trained using the same training set available but under different hyperparameters. The training hyperparameters of each model are shown in Table 3.2.

Model Number	1	2	3	4
Sample Size	612	612	612	612
Batch Size	10	10	32	32
Learning Rate	0.01	0.005	0.01	0.01
Epoch Number	11	10	15	15
Gaussian Blur Range	[0.1 - 1]	[0.1 - 1]	[0.1 - 0.5]	[0.1 - 1]
Gamma Contrast Constants	[0.25, 0.5, 1, 1.5, 2]	[0.25, 0.5, 1, 1.5]	[0.25, 0.5, 1, 1.5]	[0.25, 0.5, 1, 1.5, 2]
Training Time (hours)	21.5	21.66	19.5	19

Table 3.2: Training hyperparamaters of constituent vertebra segmentation models. A variety of hyperparameters were imposed to produce differently trained models for the final ensemble model.

Ensemble models were chosen for both the final segmentation models to promote objectivity between predictions and reduce the occurrence of misclassified pixels. The ensembles removed internal biases of the constituent models to ensure a higher correlation between the GTs and network outputs.

3.3.4 Performance Evaluation

Each model was tested on an external testing dataset as referred to in Equation 3.1. Their performances were evaluated using metrics computed with a confusion matrix, as visualised in Figure 3.8.

	PREDICTED NEGATIVE	PREDICTED POSITIVE
ACTUAL NEGATIVE	a	b
ACTUAL POSITIVE	c	d

Figure 3.8: Schematic of a confusion matrix, adapted from [39]. With regards to semantic segmentation, A True Positive (TP) relates to when a pixel is correctly labelled as part of a vertebrae or spinal region, A True Negative (TN) relates to when a pixel is correctly labelled as part of the background, A False Positive (FP) relates to when a pixel is incorrectly labelled as part of a or spinal region, A False Negative (FN) relates to when a pixel is incorrectly labelled as part of the background.

The following metrics evaluated the performance of each model using the respective equations outlined [40]:

- **Balanced Accuracy Rate** - For both the vertebrae and spine segmentation models, the number of background pixels greatly outnumbered the number of segmentation pixels. By using this metric, the measured accuracy would not be affected by majority class bias (background pixels). Using the confusion matrix, it can be evaluated as such:

$$BAR = \frac{TP(FP+TN)+TN(TP+FN)}{2(TP+FN)(FP+TN)} \tag{3.2}$$

- **Dice Similarity Coefficient** - This metric evaluated the similarity between the GT and predicted segmentation mask. This metric was chosen over the Jaccard's Index as it gave more weight to the correctly classified verterbae/spinal region pixels. Using the confusion matrix, it can be evaluated as such:

$$DSC = \frac{2 \times TP}{2 \times TP + FN + FP} \tag{3.3}$$

- **Precision** - This metric evaluated the percentage of correctly classified vertebrae/spinal region pixels with respect to the the total amount of predicted vertebrae/spinal region pixels. Using the confusion matrix, it can be evaluated as such:

$$Precision = \frac{TP}{TP+FP} \tag{3.4}$$

- **Sensitivity** - This metric evaluated the percentage of correctly classified vertebrae/spinal region pixels with respect to the the total amount of actual vertebrae/spinal region pixels. Using the confusion matrix, it can be evaluated as such:

$$Sensitivity = \frac{TP}{TP+FN} \tag{3.5}$$

- $\mathbf{F}_\beta$ **Score** - This metric evaluated the performance of the model by taking a balanced approach to precision and sensitivity. For this, β was set to 1 to report the harmonic mean between both metrics. Using the confusion matrix, it can be evaluated as such:

$$F_1 = \frac{TP^2 + TP(FN+FP)}{(TP+FP)(TP+FN)} \tag{3.6}$$

3.4 Segmentation Analysis Framework

The following sections outline four fully automatic frameworks developed to allow for robust and reliable analysis of zebrafish vertebral columns using associated segmentation masks. Each segmentation mask was considered as a binary array where points of interest had a value of 1 leaving the background as 0. This fundamental was capitalised upon through the analyses frameworks. Each framework is tailored specifically to accelerate zebrafish screening for the expert geneticist. As a result, the details outlined in these sections extremely novel and the first of their kind. All measurements were validated by the expert geneticist and fellow domain experts throughout the development stages to ensure they met an expert's standard level.

3.4.1 Spine Length Quantification

The spine length was defined as the length of curved line segment following the spine contour from the most posterior part of the skull connecting with the vertebral column, to the most caudal extremity of the zebrafish spine, as previously defined in section 3.3.1 and visualised in Figure 3.5. The spine segmentation mask was reduced to its centreline contour using the scikit-image library [41]. Using the K-means clustering algorithm implemented with the scikit learn library, five centroids were isolated along the contour over 100 iterations [42, 43]. A 3^{rd} order polynomial was fitted to the centroids identified and mapped on to the spine segmentation mask. The spine length was then determined by summing the pixel-wise length of the polynomial mapped to the spine segmentation mask. In this case, only the segmented region would be taken in to account as the background was virtually ignored as it had a value of 0. These measurements were scaled into mm using the scaling factor derived in Equation 3.1. These steps are visualised in Figure 3.9. This process was automated using a framework developed in Python and all subsequent measurements were validated by the expert geneticist using ImageJ [44].

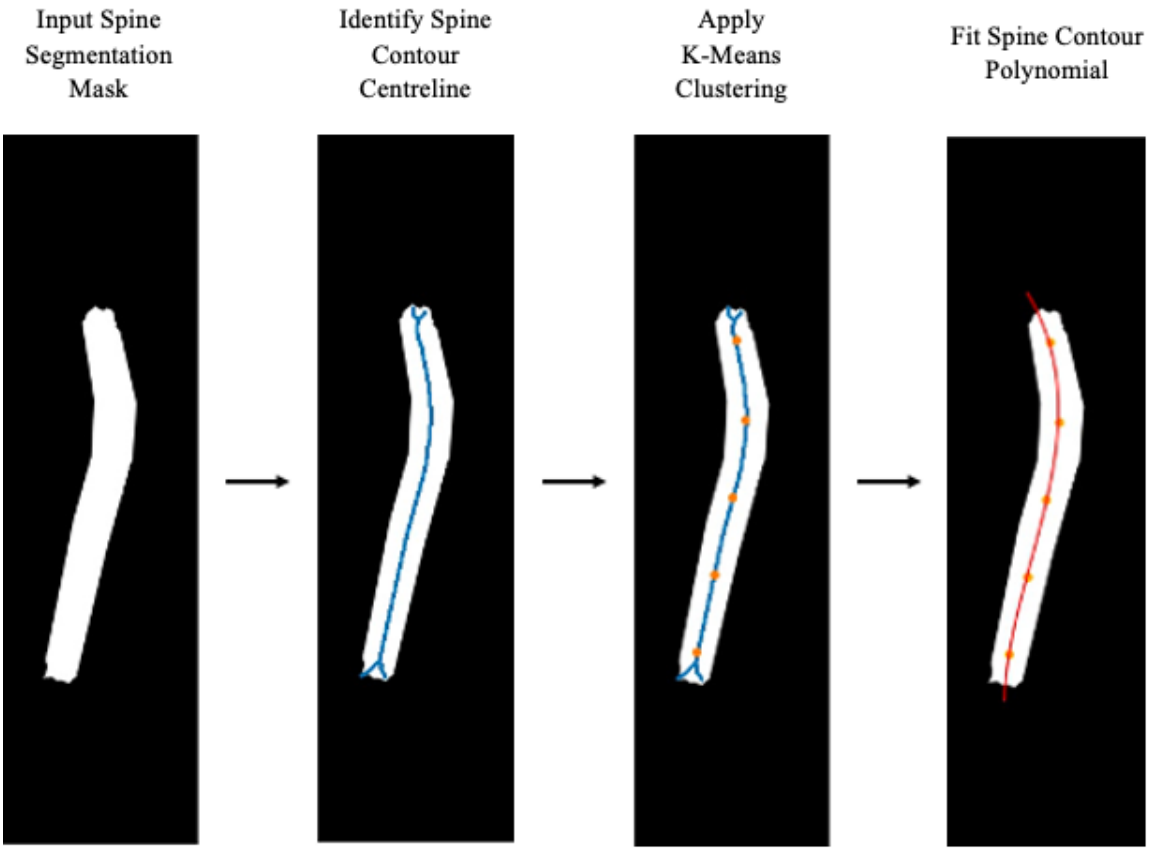


Figure 3.9: Visualisation of spine length quantification process. Spine contour (blue) isolated using the skimage library. Centroids (yellow) identified along contour using K-mean clustering. 3rd order polynomial (red) fitted to the centroids identified and interpolated onto spine segmentation mask. Spine length determined by summing the pixelwise length of mapped polynomial. This specific example was 2125 pixels long, correlating to 21.61 mm.

3.4.2 Spine Curvature Quantification

Before this study, Yushi Yang developed a framework that described the average curvature of the zebrafish spine, k, relating to the following equation for signed curvature:

$$k = \frac{y''}{(1+y'^2)^{\frac{3}{2}}} \tag{3.7}$$

This allowed for curvature visualisations and measurements as shown in Figure 3.10. This method was sufficient for simple analysis or a brief overview of spine curvature. However it was not a solid framework that could adequately identify, position and measure multiple spine curvatures in a given sample.

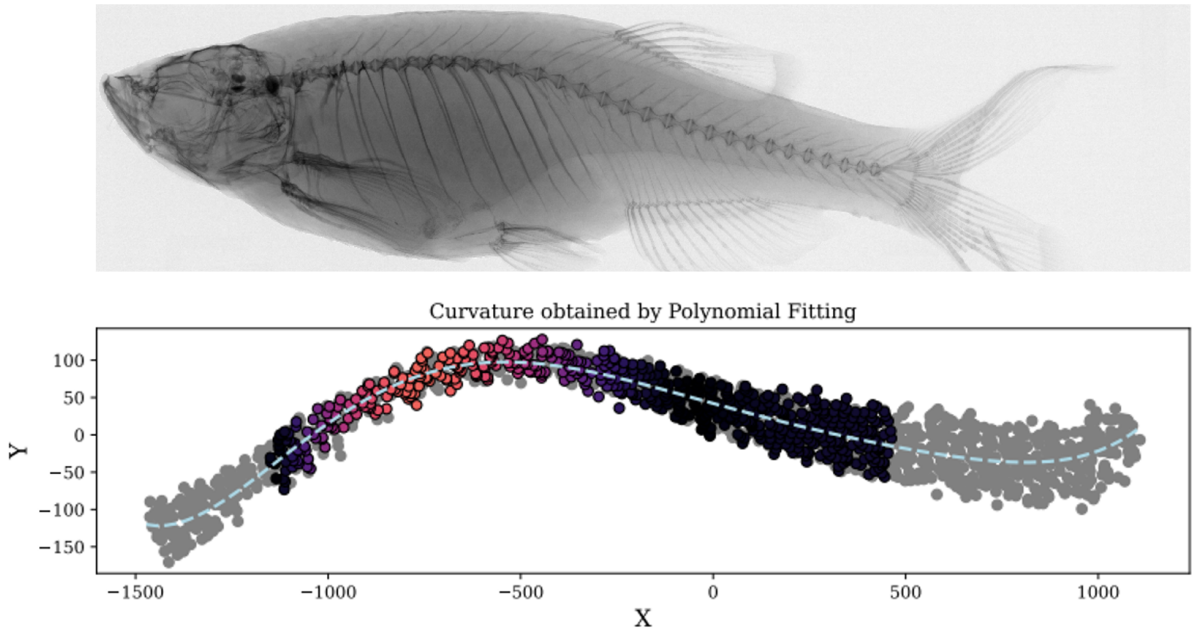


Figure 3.10: Visualisation the results generated from the original spine curvature quantification framework previously in place. Average sample curvature - 0.000275 unit^{-1}, maximum sample curvature - 0.000818 unit^{-1}, standard deviation of curvature - 0.000267 unit^{-1}. The brighter the colour of the representative curvature polynomial, the more curved the particular section.

The Cobb Angle method, as visualised in Figure 2.6, defined human spine curvature as the angle made between the most tilted vertebra endplates on X-rays. Tu *et al.* outlined an approach of measuring human spinal curvature based on the tangents of a 6$^{\text{th}}$ order polynomial fitted to the spine contour [23]. This yielded highly accurate results, however could not be used in a clinical setting as it did not directly reference the clinical standard Cobb Angle. Tu *et al.*'s method of quantifying spine curvature can be modified so that it can relate to the Cobb Angle method more and be further applied to the zebrafish spine. Where the most tilted vertebra endplates should for the Cobb Angle should occur in close proximity to the position of the steepest slopes or inflection points, I_{n}, of Tu *et al.*'s polynomial, and the extreme points, E, should be close to the extreme spine curvature. By relating the Cobb Angle method to the zebrafish, the angle of spine curvature was defined as the angle $|< I_1OI_2|$ and its location - referring to the anterior/posterior of the fish - could be determined by using the position of E, as shown in Figure 3.11.

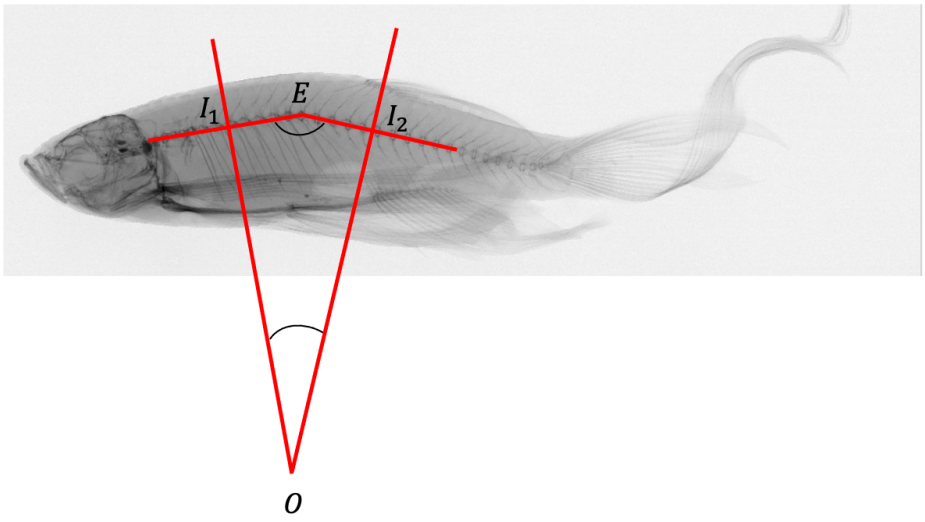


Figure 3.11: Visualisation of Cobb Angle method originally used for humans being applied to the zebrafish. The angle of curvature for the zebrafish spine is defined as $|<I_1OI_2|$.

Where I_n indicated the points of inflection and E the extreme points of the 3rd order polynomial from subsection 3.4.1. These points were identified using the PyTorch library [35]. The distances between each point were defined using Euclidean distance, defined in Equation 3.8:

$$|I_1E| = \sqrt{\left(x_E - x_{I_1}\right)^2 + \left(y_E - y_{I_1}\right)^2} \tag{3.8}$$

Angles $|<I_1EI_2|$ and $|<I_1OI_2|$ were then determined using the following equations:

$$|<I_1EI_2| = \cos^{-1}\left(\frac{|I_1E|^2 + |EI_2|^2 - |I_1I_2|^2}{2|I_1E||EI_2|}\right) \tag{3.9}$$

$$|<I_1OI_2| = 180^\circ - |<I_1EI_2| \tag{3.10}$$

This task was automated and the process of governing the framework are visualised in Figure 3.12. All results generated by this automatic framework were validated by the expert geneticist and tested against the previous spine curvature framework.

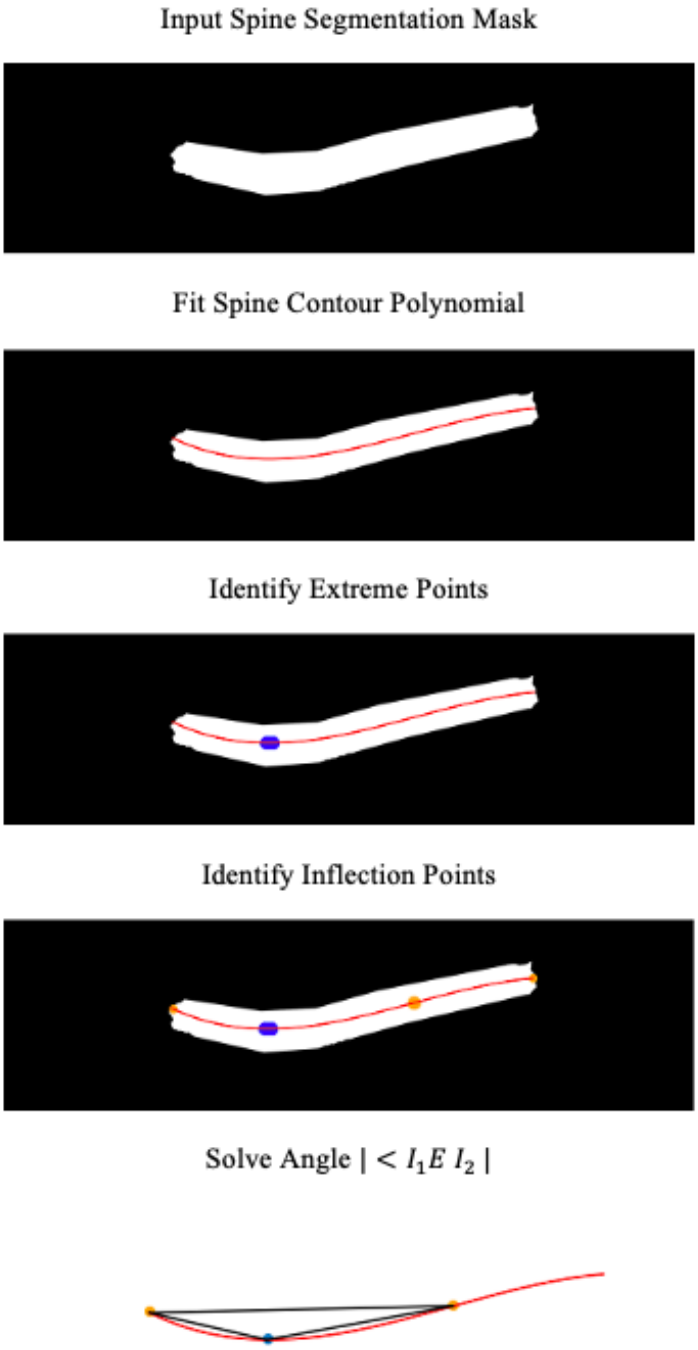


Figure 3.12: Visualisation of spine curvature quantification process. Spine segmentation mask is introduced into automatic framework and the fitted spine contour polynomial (red) is defined. Extreme points (blue) are identified by computing the first derivative of the spine contour polynomial. Inflection points (yellow) are identified by computing the second derivative of the spine contour polynomial. $| < I_1 E I_2 |$ is determined using Cosine Rule.

3.4.3 Vertebra Area Quantification

The area for each individual vertebra was calculated pixel-wise by summing vertebral regions using bounding boxes generated with the scikit-image library in Python [41]. Background regions were ignored due to their value of 0, hence by summing the region defined by the bounding box only the vertebra were included. This is visualised in Figure 3.13. These measurements were then further scaled into mm^2 using the scaling factor derived in Equation 3.1. To automate the process, a framework was developed to calculate the area of each segmented vertebra mask from head to tail. The framework also provided the respective bounding box dimensions for each subsequent segment. All results generated by this automatic framework were validated by the expert geneticist using ImageJ [44].

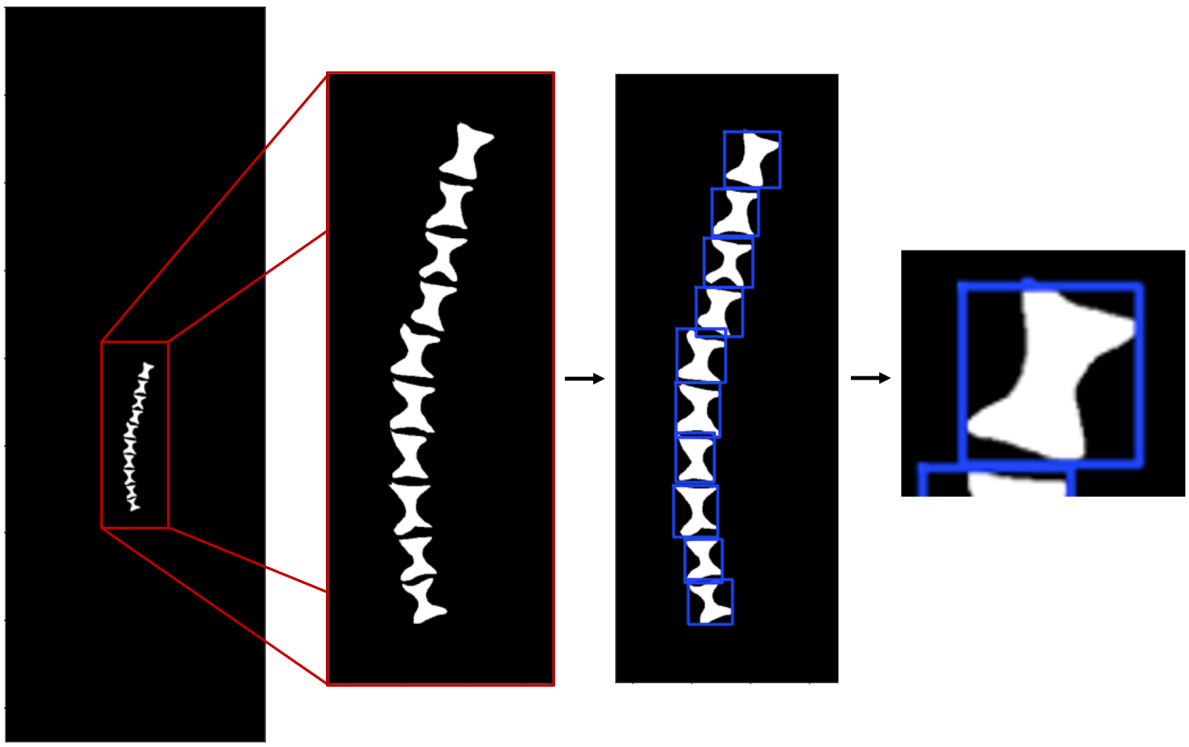


Figure 3.13: Visualisation of vertebra area quantification process. Vertebrae segmentation mask introduced into automatic framework. Bounding boxes (blue) are defined using the scikit-image library in Python. Vertebra area is quantified pixel-wise by summing the respective bounding box sections. This specific example was 3953 $pixels^2$, correlating to 0.41 mm^2.

3.4.4 Vertebra Length Quantification

The length of the vertebra was defined as the line segment parallel to the spine contour. Using the K-means clustering algorithm of the scikit learn library, one centroid cluster for each vertebra segment was isolated over 100 iterations [42, 43]. A 2^{nd} order polynomial was fitted to these centroid points and subsequently mapped to the verterbae segmentation mask. The mask was sectioned into smaller segments using the bounding box dimensions of subsection 3.4.3 to isolate the individual vertebra length line segments. The vertebra length was then determined as the Euclidian distance (Equation 3.8) between the start and end point of the line segment. The measurements were then further scaled into mm using the scaling factor derived in Equation 3.1. These steps are visualised in Figure 3.14. This process was automated using a framework developed in Python and all subsequent measurements were validated by the expert geneticist using ImageJ [44].

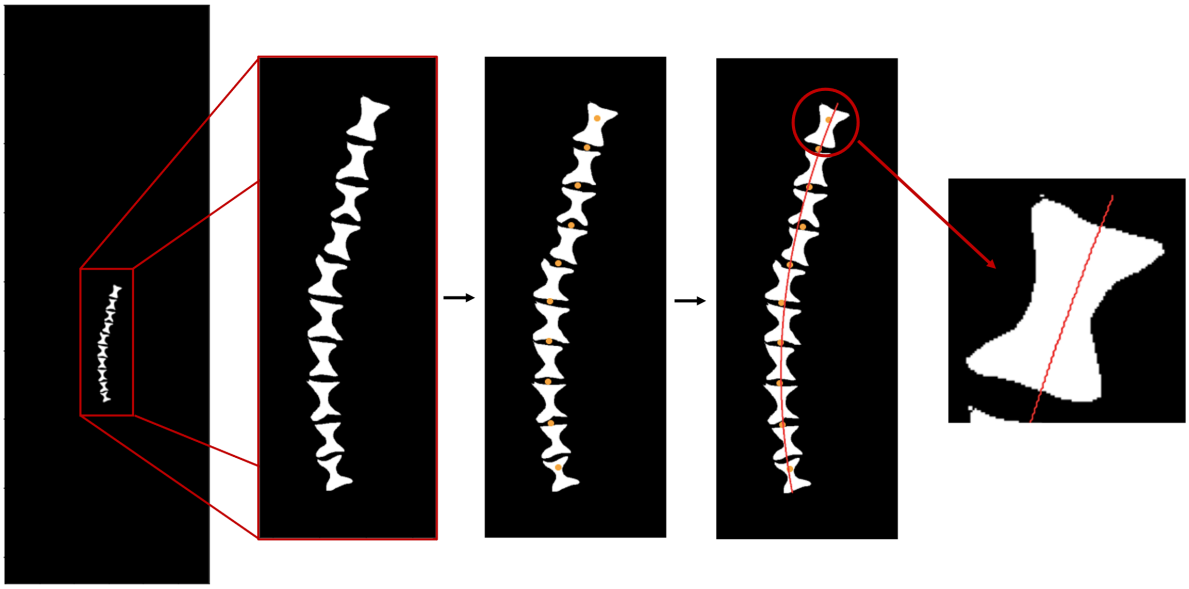


Figure 3.14: Visualisation of vertebra length quantification process. Vertebrae segmentation mask introduced into automatic framework. Approximate centroids (yellow) identified for each vertebra segmentation using K-mean clustering. 2^{nd} order polynomial (red) fitted to the centroids identified and interpolated onto vertebrae segmentation mask. Vertebra length subsequently determined by computing the Euclidean distance between the start and end point of vertebra segments defined by the fitted polynomial. This specific example was 81.93 pixels long, correlating to 0.83 mm.

3.5 Summary

Within this chapter, the processes outlined in each individual automatic framework of the whole novel pipeline are the first of their kind. There is no previous literature that has successfully combined the genetic screening of therapeutic targets with processes involving zebrafish and AI to design a fully automatic pipeline. The results of each section outlined in this chapter are displayed in the following chapter.

CHAPTER

FOUR

RESULTS

4.1 Introduction

This chapter outlines the results of each framework composing the novel pipeline proposed in chapter 3. The training and validation progressions of the segmentation models developed previously are displayed. The results of their performances based on 100 samples from the external testing dataset are outlined. 10 random samples from the testing dataset were extracted and used to assess the validity and reliability of the four fully automatic frameworks developed in section 3.4. This allowed 10 spines and 100 vertebrae to be assessed for correlations between manual measurements performed by domain experts and the automatic measurements generated by associated frameworks designed for the analysis of zebrafish phenotypes. Results are further analysed in-depth in chapter 5.

4.2 Automatic Spine Segmentation

4.2.1 U-Net Training and Validation

Figure 4.1 visualises the learning curves of the four constituent models during training that were later used to generate the final spine segmentation ensemble model. Each plot visualises the training and validation losses reported after each epoch during training stages.

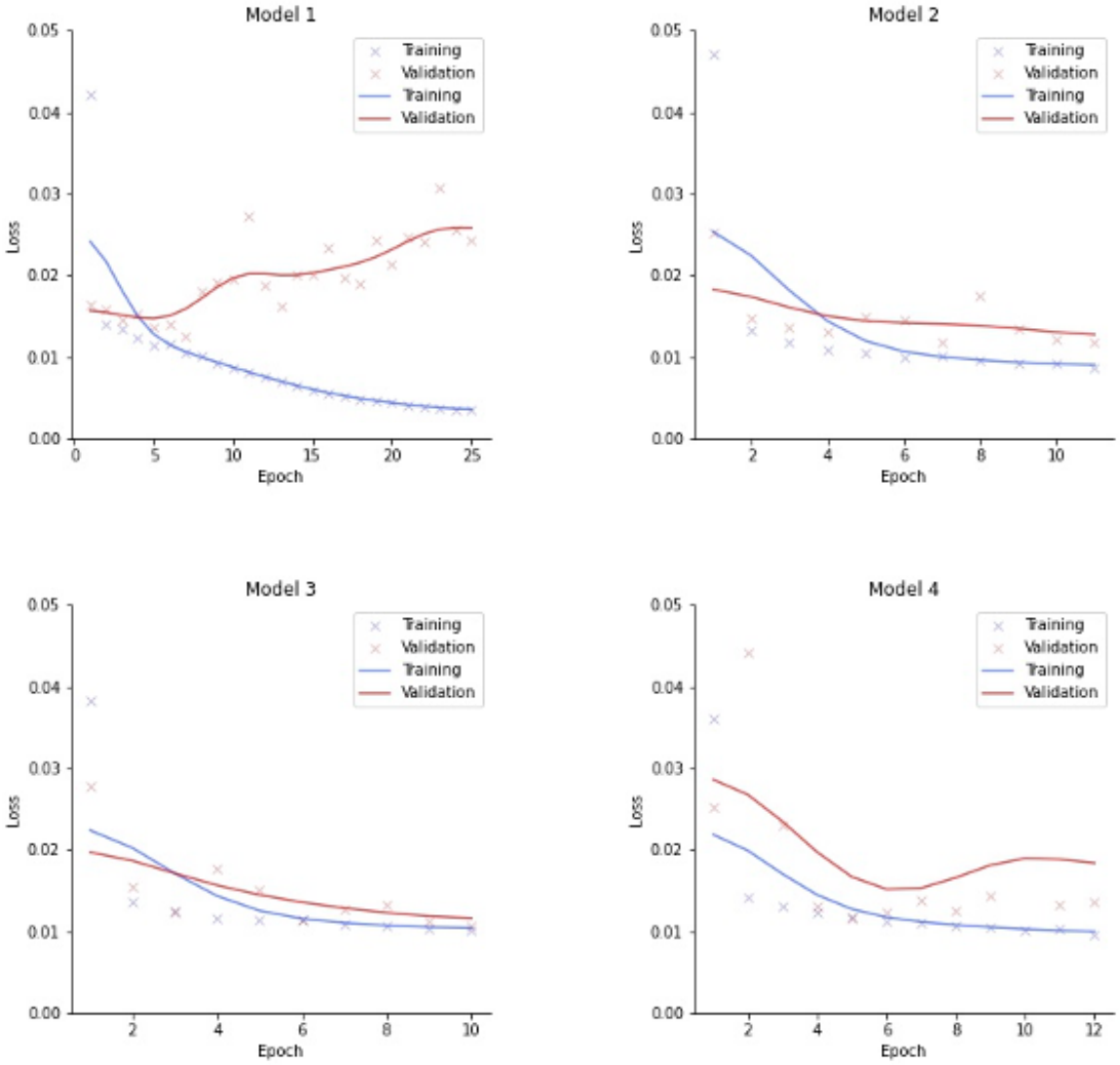


Figure 4.1: Training and validation losses of constituent spine segmentation models. Blue crosses indicate the true values of training losses, red crosses indicate the true values of validation losses, blue curve indicates the Gaussian filtered trend of training losses, red curve indicates the Gaussian filtered trend of validation losses.

4.2.2 Comparison with Ground Truth

Constituent Models

Figures 4.2 to 4.5 visualise the performances of the constituent models when tested on three random samples from the external testing dataset. For each sample tested, the inputted zebrafish X-ray sample is depicted alongside its associated manually annotated GT. The model's raw output is depicted as Network Output (NO), subsequently binarized - such that negative values were 0 and positive were 1 - and depicted as Binary Output (BO).

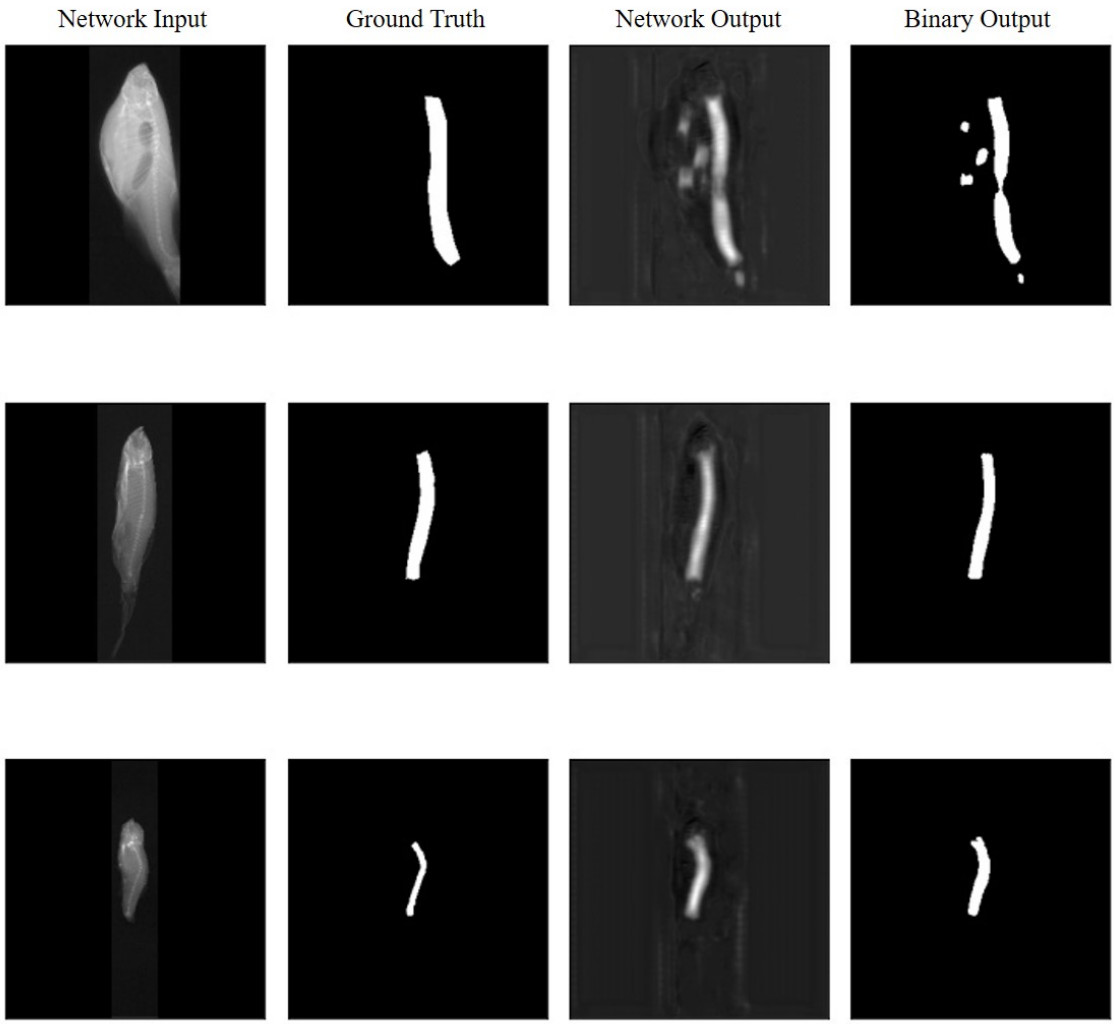


Figure 4.2: Performance of spine segmentation model 1 vs. Ground Truths. Samples shown were chosen at random.

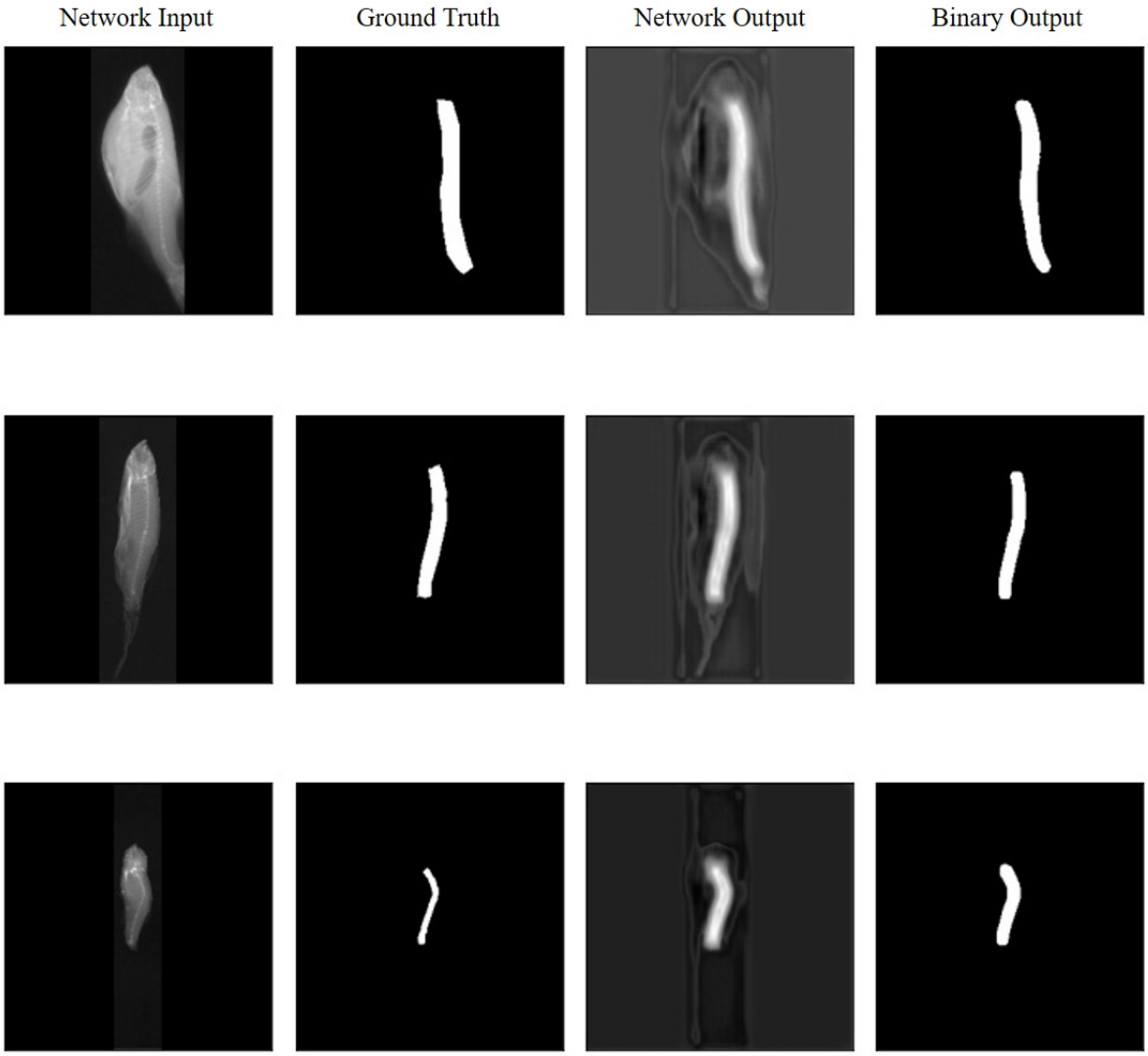


Figure 4.3: Performance of spine segmentation model 2 vs. Ground Truths. Samples shown were chosen at random.

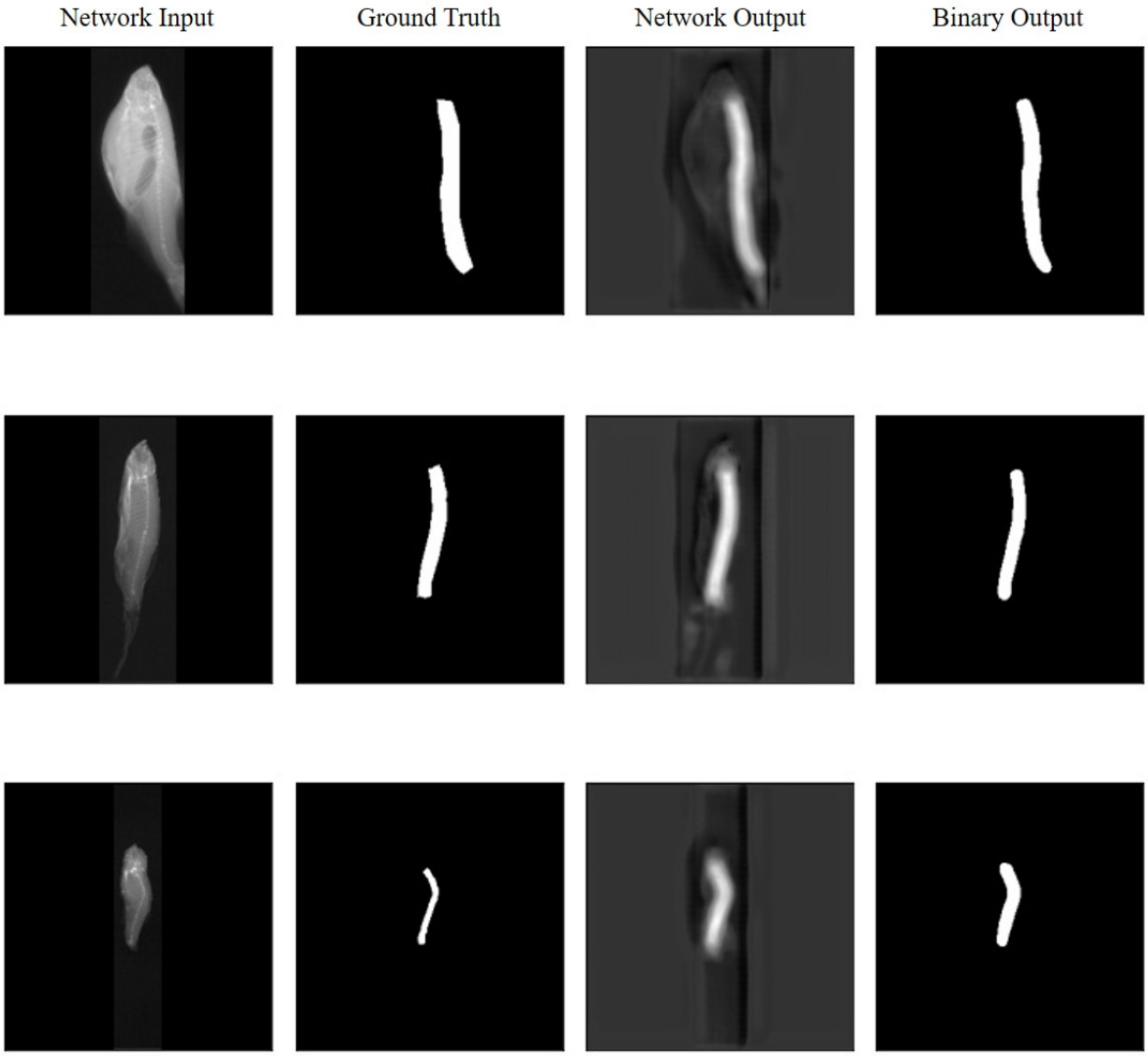


Figure 4.4: Performance of spine segmentation model 3 vs. Ground Truths. Samples shown were chosen at random.

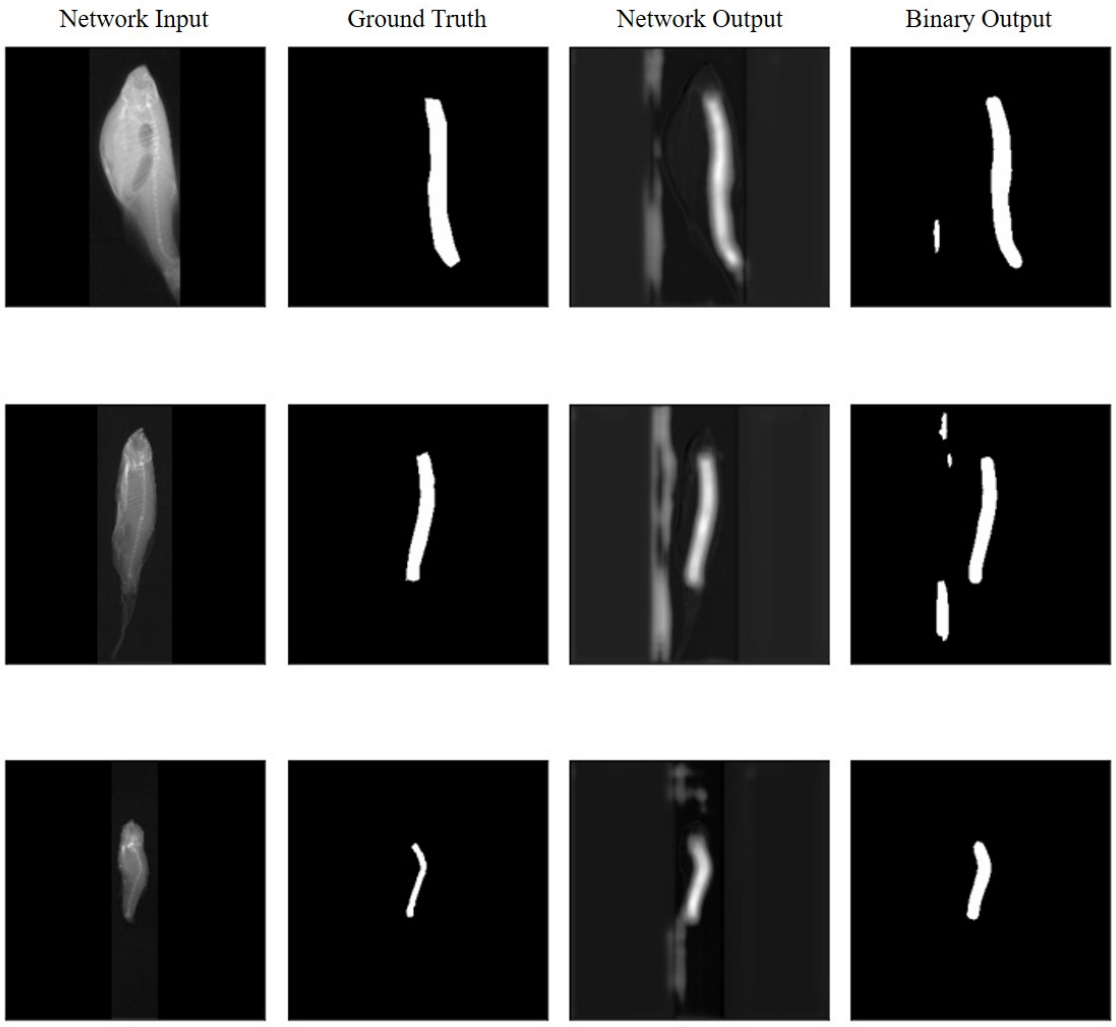


Figure 4.5: Performance of spine segmentation model 4 vs. Ground Truths. Samples shown were chosen at random.

All models took on average 5 minutes to run on all the external testing data, segmenting the spine of each sample in approximately 3 seconds.

Spine Segmentation Ensemble Model

The final spine segmentation ensemble model was based on the average BO of all four constituent models. A pixel was classified either "spine" or "background" when more than 50% of the models classified the pixel as "spine" or "background" respectively. Figure 4.6 visualises the performance of the spine segmentation ensemble model when tested on the three random samples used in Figures 4.2 to 4.5. For each sample tested, the specific X-ray sample, associated GT, average BOs of the constituent models and ensemble output are visualised.

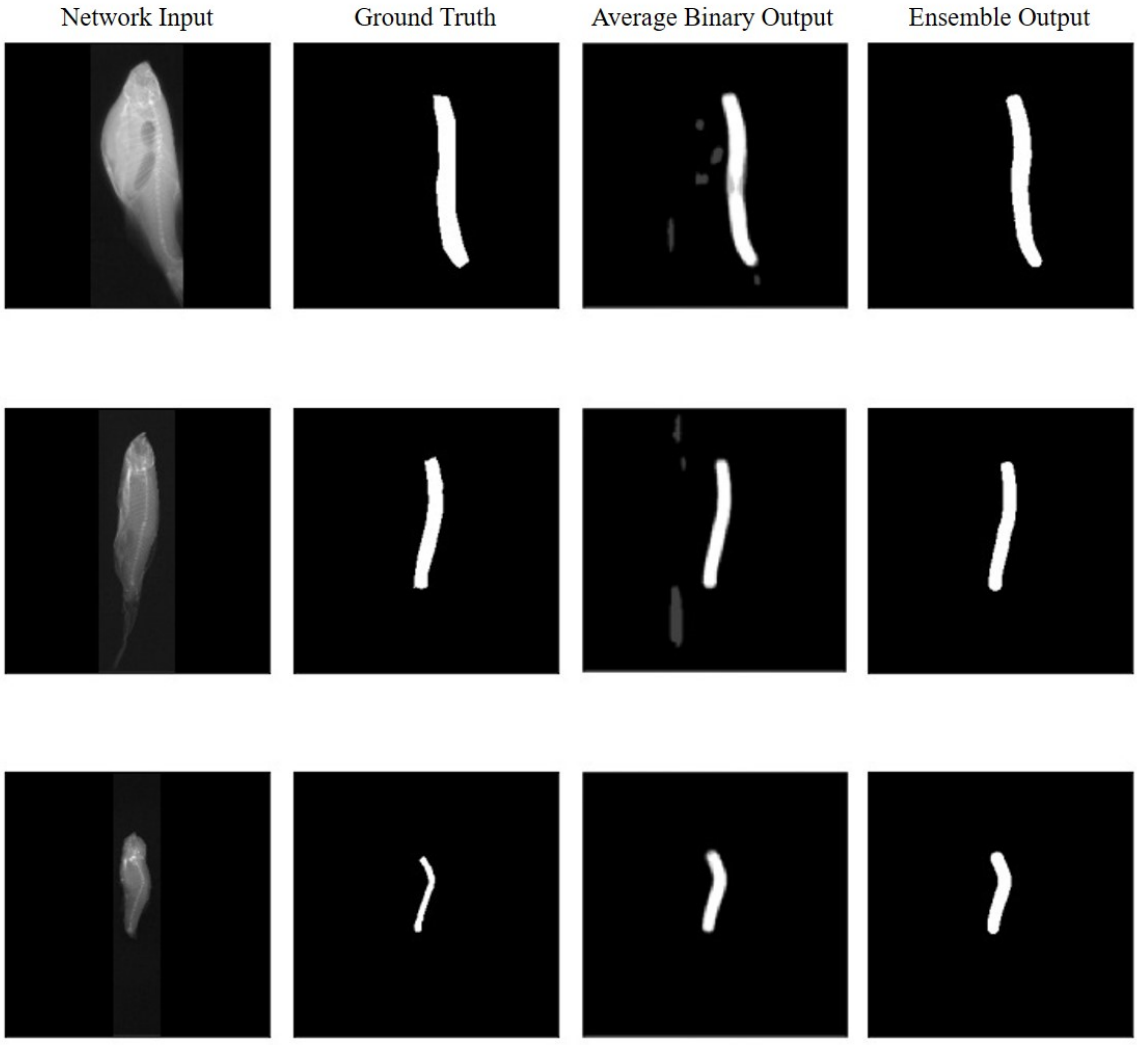


Figure 4.6: Performance of spine segmentation ensemble model. Final output of the ensemble was based on the majority vote of the constituent spine segmentation models.

4.2.3 Performance Evaluation

The performances of all four constituent models and the spine segmentation ensemble model were evaluated using the external testing dataset samples. Performances were evaluated using the metrics as outlined in subsection 3.3.4 and results are shown in Table 4.1.

Model	BAR	DSC	F1 Scores	Precision	Sensitivity
1	0.92	0.89	0.98	0.96	0.83
2	0.92	0.90	0.98	0.96	0.84
3	0.94	0.92	**0.98**	**0.96**	**0.97**
4	**0.95**	0.91	0.96	0.92	0.90
Final Ensemble	0.94	**0.92**	0.98	0.96	0.90

Table 4.1: Summary of the performance evaluations of the spine segmentation models. The highest performing metrics are outlined in bold.

The final spine segmentation ensemble model achieved the highest Dice Similarity Coefficient across all other constituent models. A 95% Confidence Interval was determined for this evaluation metric to four significant figures. The Dice Similarity Coefficient for the final spine segmentation ensemble model was above this interval.

All models were extremely high performing and well-trained. As a result, the spine segmentation ensemble model gives an average performance across all constituent models.

4.3 Automatic Vertebrae Segmentation

4.3.1 U-Net Training and Validation

Figure 4.7 visualises the learning curves of the four constituent models during training that were later used to generate the final automatic vertebrae segmentation ensemble model. Each plot visualises the training and validation losses reported after each epoch during training stages.

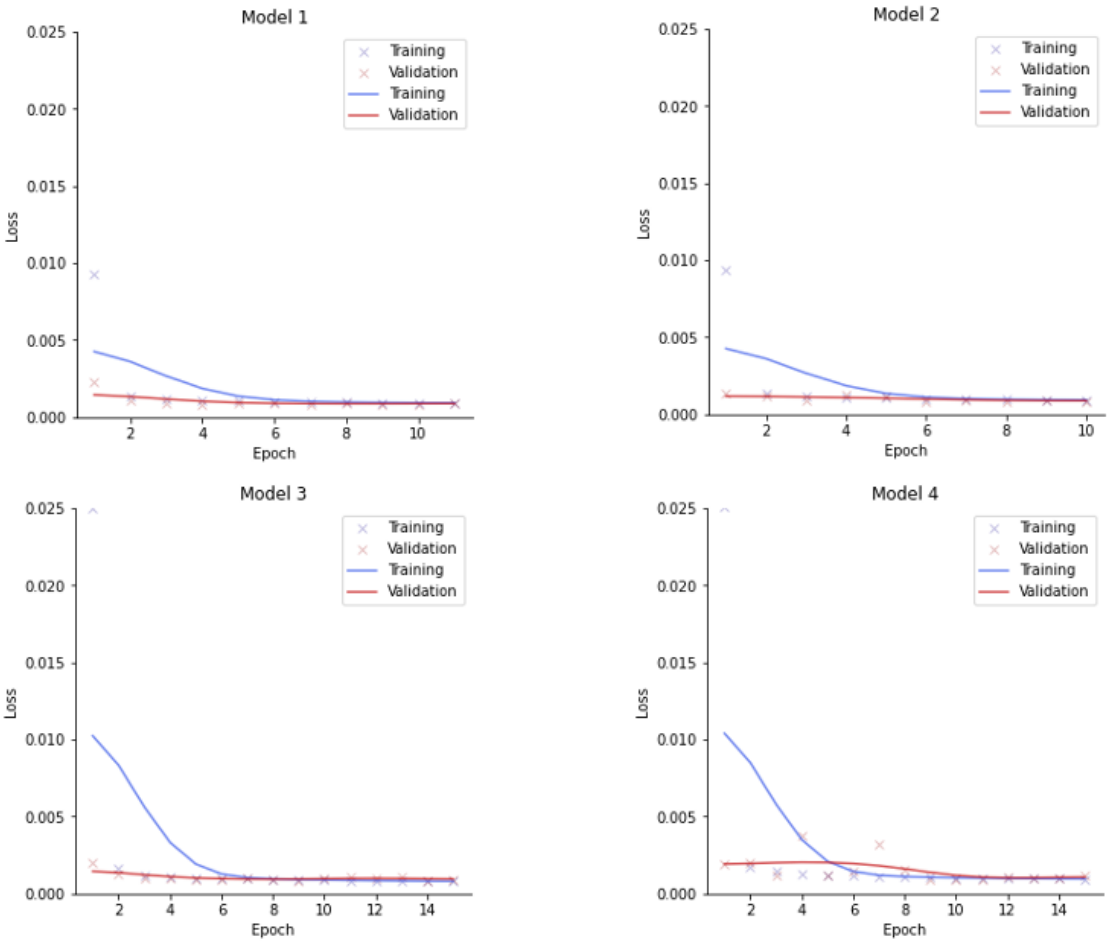


Figure 4.7: Training and validation losses of constituent vertebrae segmentation models. Blue crosses indicate the true values of training losses, red crosses indicate the true values of validation losses, blue curve indicates the Gaussian filtered trend of training losses, red curve indicates the Gaussian filtered trend of validation losses

4.3.2 Comparison with Ground Truth

Constituent Models

Figures 4.8 to 4.11 visualise the performances of the constituent models when tested on three random samples from the external testing dataset. For each sample tested, the inputted zebrafish X-ray sample is depicted alongside its associated manually annotated GT. The model's raw output is depicted as NO, subsequently binarized - such that negative values were 0 and positive were 1 - and depicted in BO.

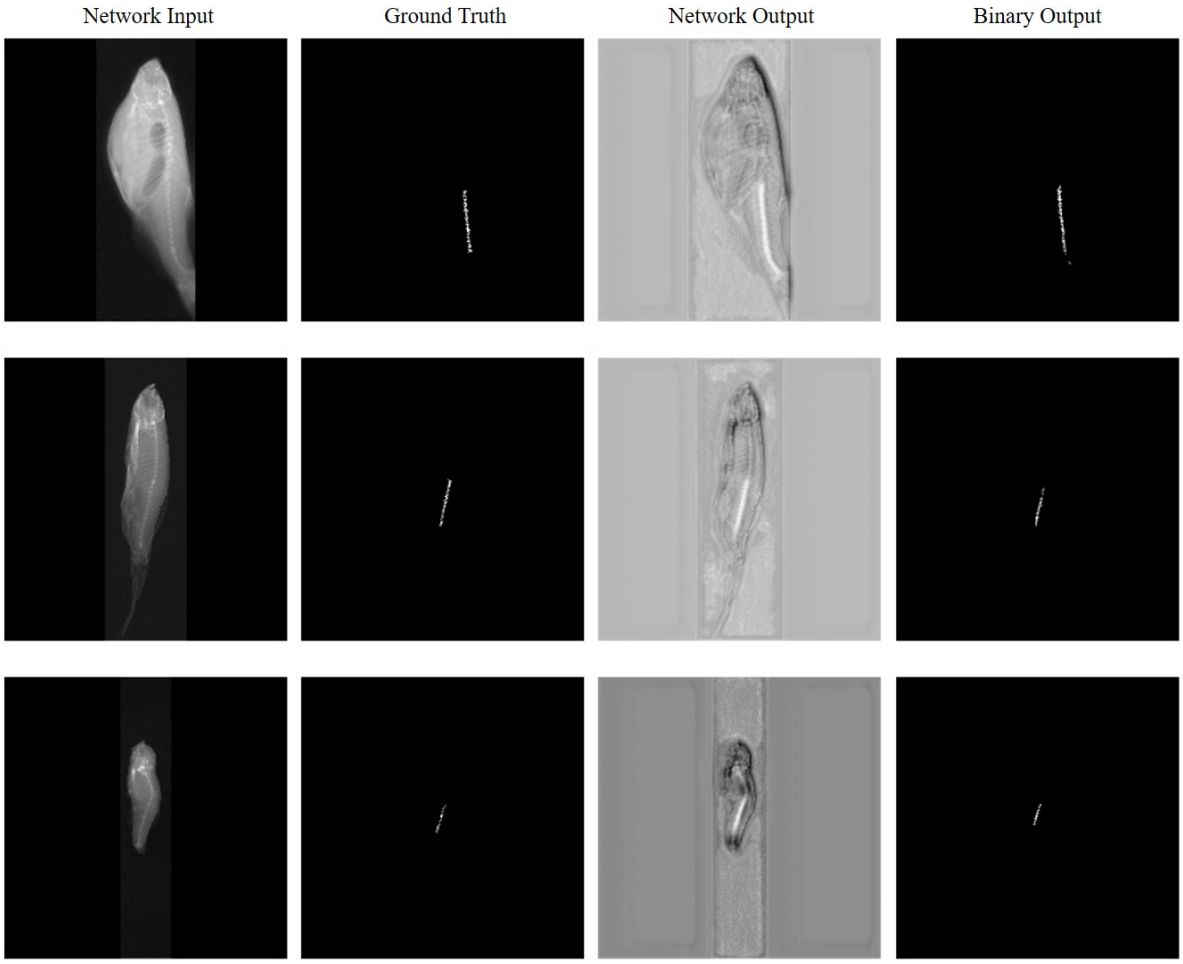


Figure 4.8: Performance of vertebrae segmentation model 1 vs. Ground Truths. Samples shown were chosen at random.

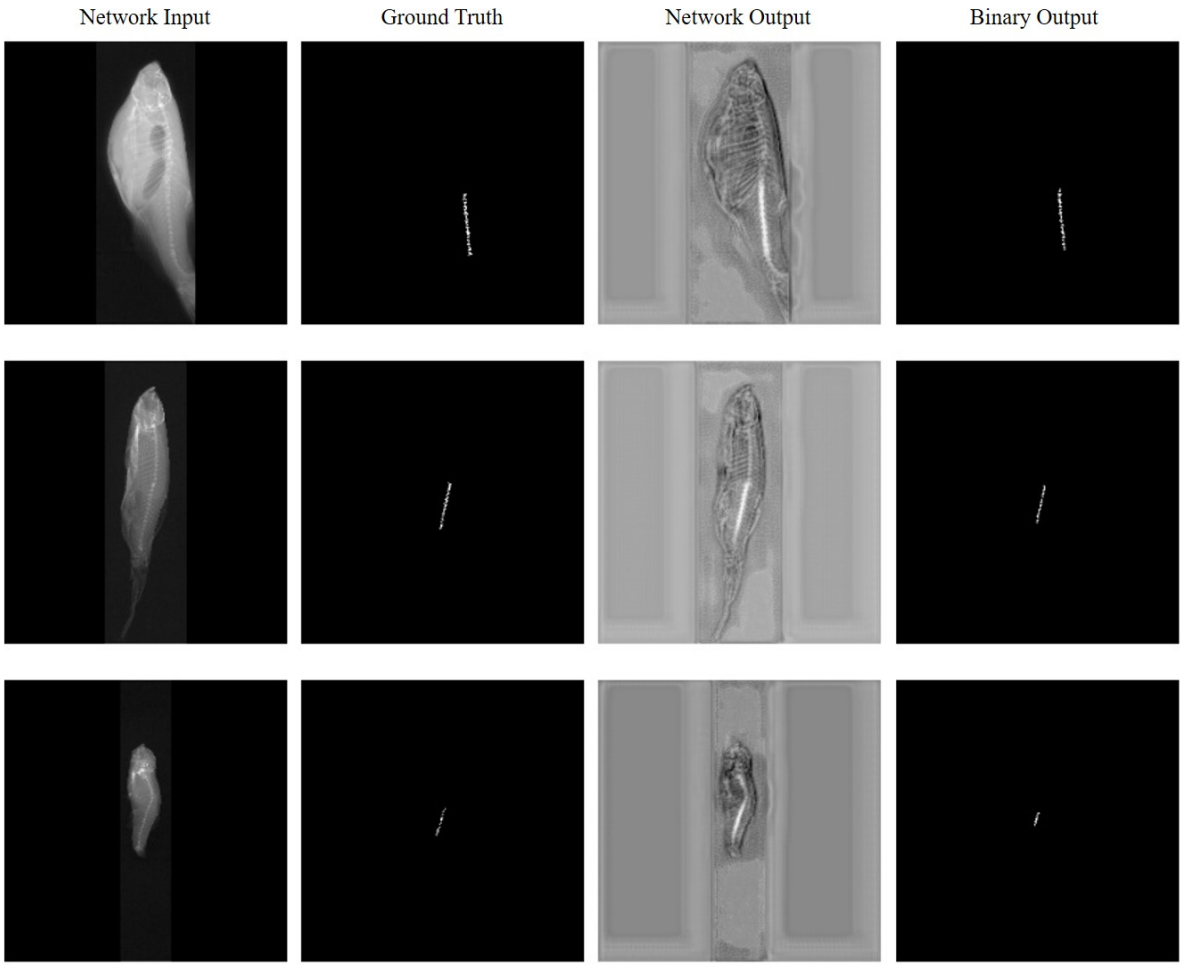


Figure 4.9: Performance of vertebrae segmentation model 2 vs. Ground Truths. Samples shown were chosen at random.

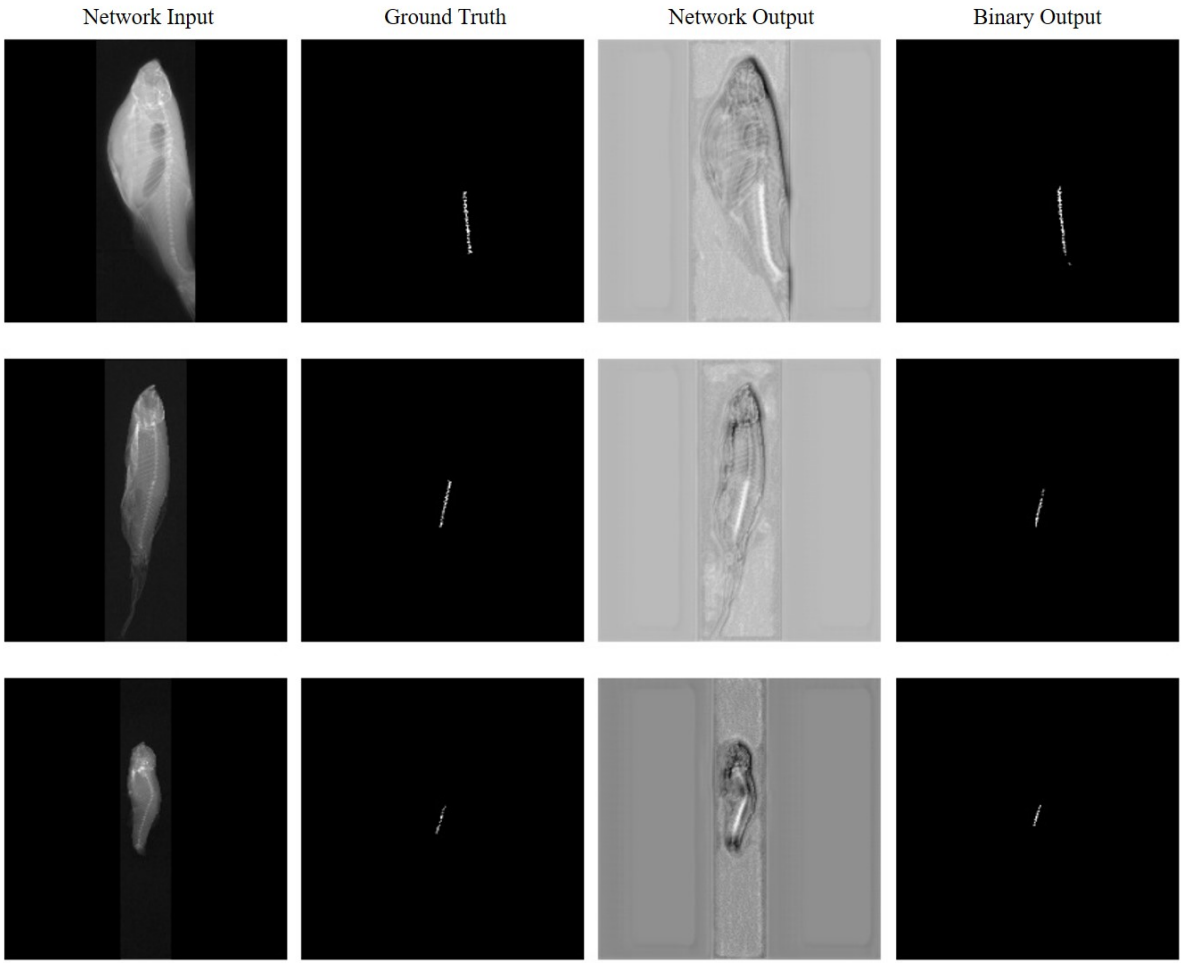


Figure 4.10: Performance of vertebrae segmentation model 3 vs. Ground Truths. Samples shown were chosen at random.

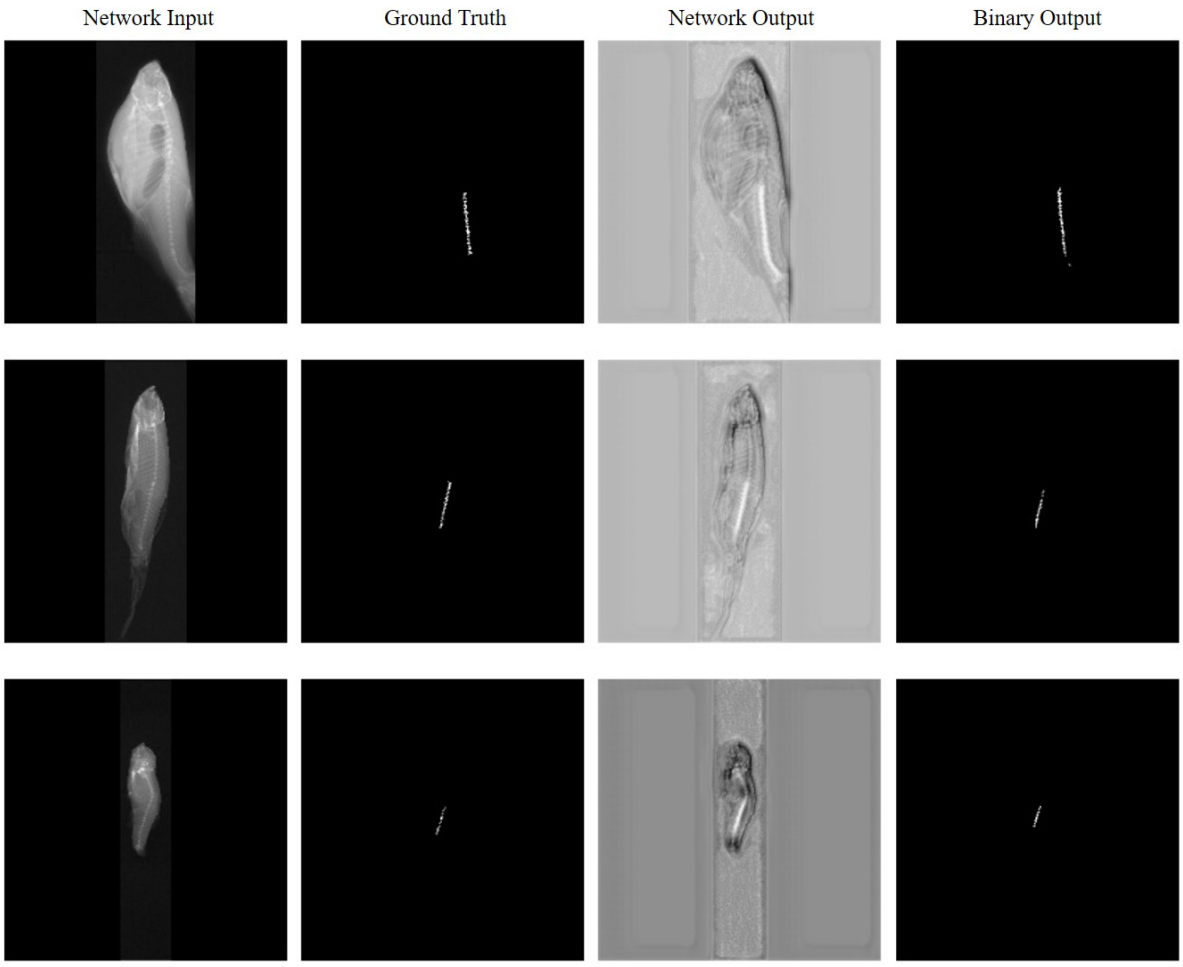


Figure 4.11: Performance of vertebrae segmentation model 4 vs. Ground Truths. Samples shown were chosen at random.

On average, each model took 5 minutes to test the entire external testing dataset. This lead to an average time of 3 seconds for the generation of each vertebrae segmentation mask.

Vertebrae Segmentation Ensemble Model

The final vertebrae segmentation ensemble model was based on the average BO of all four constituent models. A pixel was classified either "vertebra" or "background" when more than 50% of the models classified the pixel as "vertebra" or "background" respectively. Figure 4.12 visualises the performance of the vertebrae segmentation ensemble model when tested on the three random samples used in Figures 4.8 to 4.11. For each sample tested, the specific X-ray sample, associated GT, average BOs of the constituent models and ensemble output are visualised.

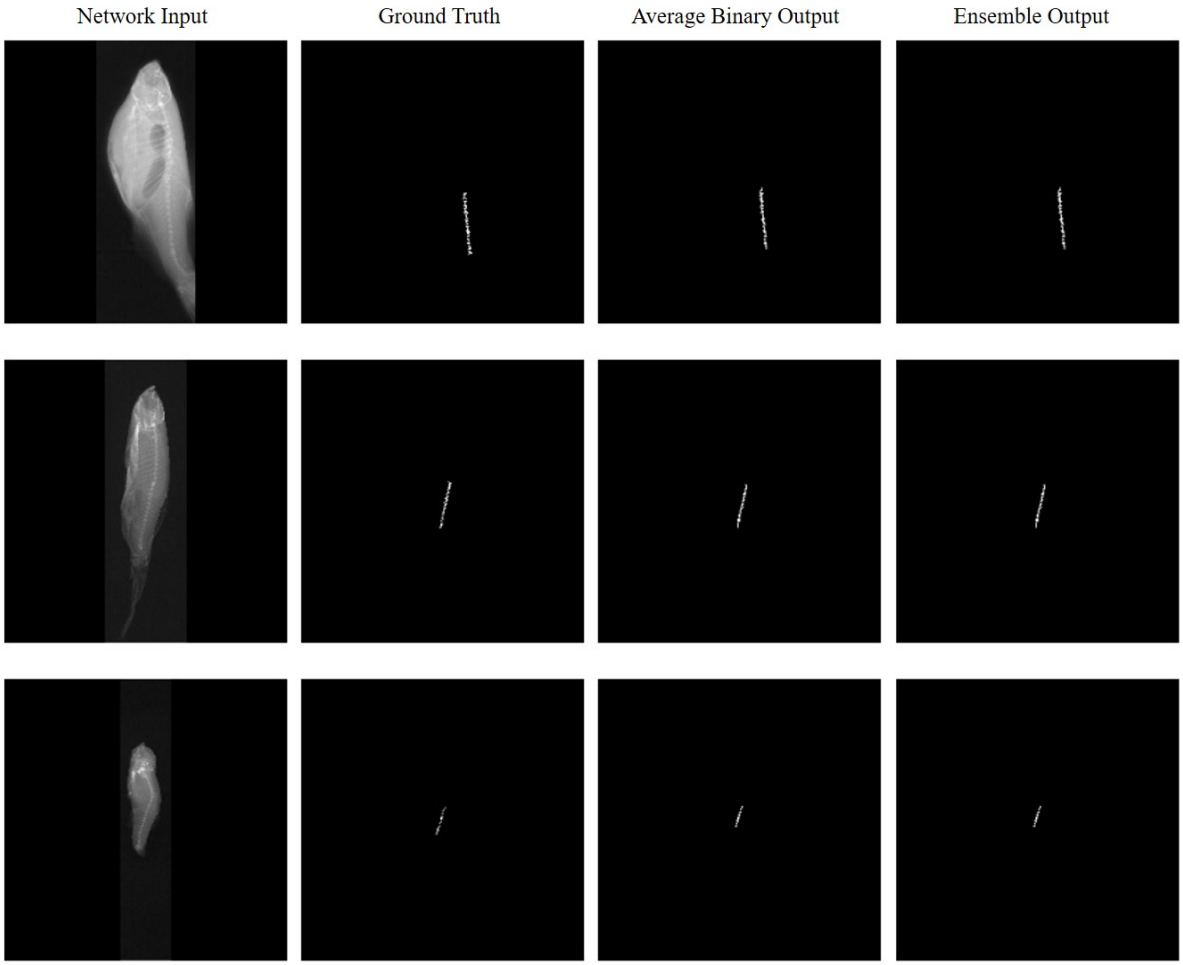


Figure 4.12: Performance of final vertebrae segmentation ensemble model. Final output of the ensemble was based on the majority vote of the constituent vertebrae segmentation models.

4.3.3 Performance Evaluation

The performances of all four constituent models and the vertebrae segmentation ensemble model were evaluated using the external testing dataset samples. Performances were evaluated using the metrics as outlined in subsection 3.3.4 and results are shown in Table 4.2.

Model	BAR	DSC	F1 Scores	Precision	Sensitivity
1	0.89	0.77	0.87	0.77	0.77
2	0.87	0.77	**0.89**	**0.80**	0.74
3	0.88	0.77	0.87	0.78	0.76
4	0.86	0.74	0.87	0.77	0.72
Final Ensemble	**0.90**	**0.79**	0.87	0.77	**0.80**

Table 4.2: Summary of the performance evaluations of the vertebrae segmentation models. The highest performing metrics are outlined in bold.

The final vertebrae segmentation ensemble model achieved the highest Balanced Accuracy Rate, Dice Similarity Coefficient and Sensitivity scores across all other constituent models. A 95% Confidence Interval was determined for these evaluation metrics to four significant figures. The Balanced Accuracy Rate, Dice Similarity Coefficient and Sensitivity for the final vertebrae segmentation ensemble model were above each respective interval.

These models were not efficiently trained subsequently resulting in the results outlined. The verterbae segmentation model was able to isolate the common outputs of the constituent vertebrae segmentation models, ignore their internal biases and perform well when compared to the GTs.

4.4 Segmentation Analysis Framework

4.4.1 Spine Length Quantification

Appropriate spine length quantification relied on adequate fitting of the 3rd order spine contour polynomial relating to the spine centreline. Using the external testing dataset, the spine contour polynomial was fit to the spine centreline with a mean Pearson Correlation of 0.97 and a standard deviation of 0.05.

Figure 4.13 shows the linear correlation between the manual measurements performed by domain experts and the automatic measurements performed by the spine length quantification framework of subsection 3.4.1. The framework achieved a Pearson Correlation of 0.97 and an MAE of 123.78 pixels or 1.26 mm. Using a Student T-test, there was no significant statistical difference determined between the two measurement methods. These results are summarised in Table 4.3.

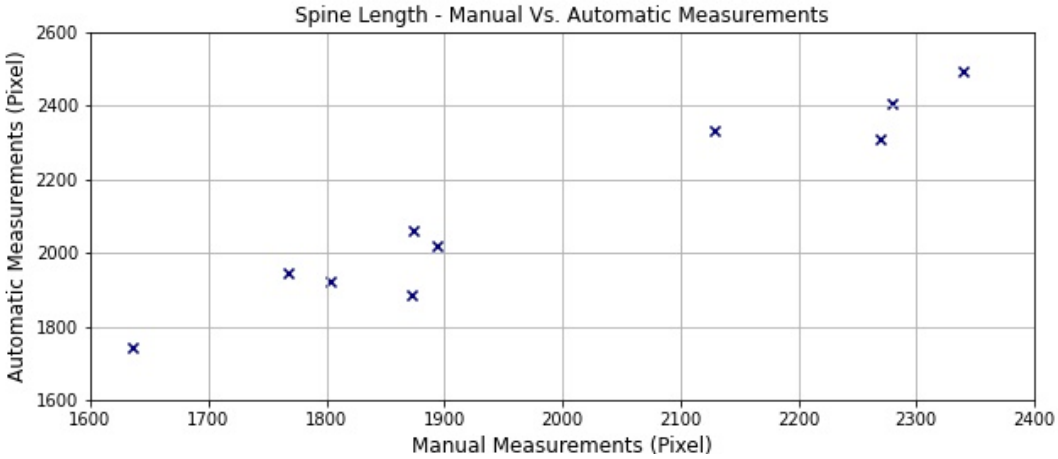


Figure 4.13: Measurement correlation between automated and manual methods of spine length quantification.

Pearson Correlation	Mean Absolute Error		p-value
0.97	123.78 pixels	1.26 mm	0.2840

Table 4.3: Summary of automatic vs. manual measurements for spine length quantification.

4.4.2 Spine Curvature Quantification

Figure 4.14 shows the linear correlation between the manual measurements performed by the framework in place originally and the automatic measurements performed by the spine curvature quantification framework of subsection 3.4.2. As these frameworks imposed a different resulting measurements ($pixel^{-1}$ vs. degree of curvature), the measurements were standardised before being displayed using the scikit learn library [43]. The framework achieved a Pearson Correlation of 0.64.

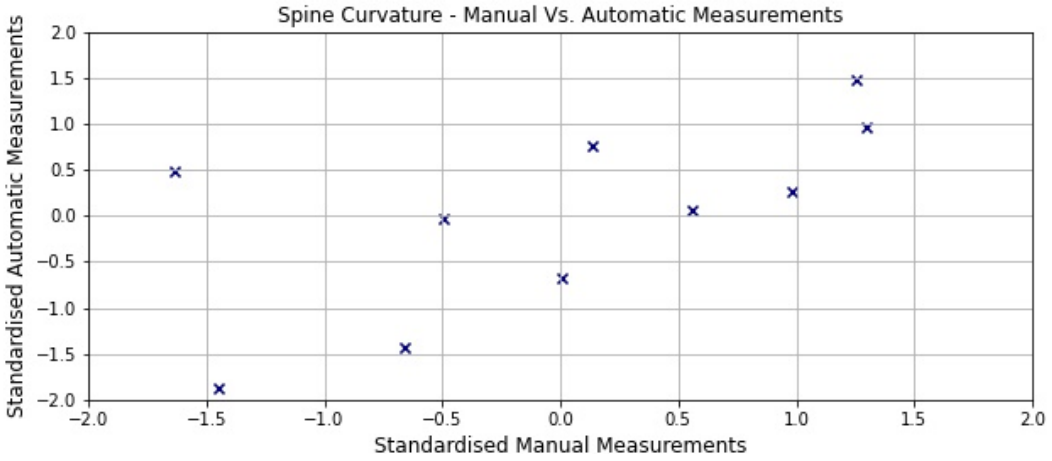


Figure 4.14: Measurement correlation between automated and manual methods of spine curvature quantification.

4.4.3 Vertebra Area Quantification

Figure 4.15 shows the linear correlation between the manual measurements performed by domain experts and the automatic measurements performed by the vertebra area quantification framework of subsection 3.4.3. The framework achieved a Pearson Correlation of 0.92 and an MAE of 190.39 pixels^2 or 0.02 mm^2. Using a Student T-test, there was no significant statistical difference determined between the two measurement methods. These results are summarised in Table 4.4.

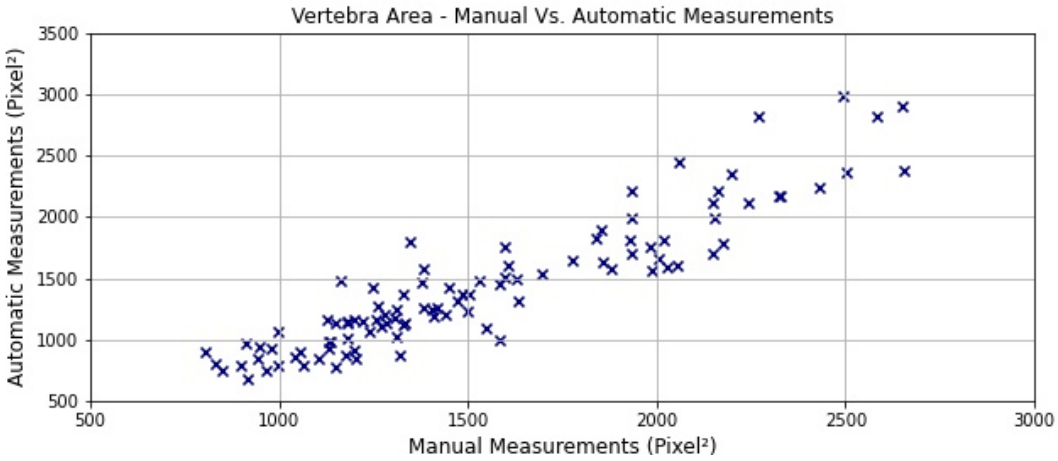


Figure 4.15: Measurement correlation between automated and manual methods of vertebra area quantification.

Pearson Correlation	Mean Absolute Error		p-value
0.92	190.39 pixels^2	0.02 mm^2	0.1361

Table 4.4: Summary of automatic vs. manual measurements for vertebra area quantification.

4.4.4 Vertebra Length Quantification

Figure 4.16 shows the linear correlation between the manual measurements performed by domain experts and the automatic measurements performed by the vertebra length quantification framework of subsection 3.4.4. The framework achieved a Pearson Correlation of 0.85 and an MAE of 5.03 pixels or 0.05 mm. Using a Student T-test, there was no significant statistical difference determined between the two measurement methods. These results are summarised in Table 4.5.

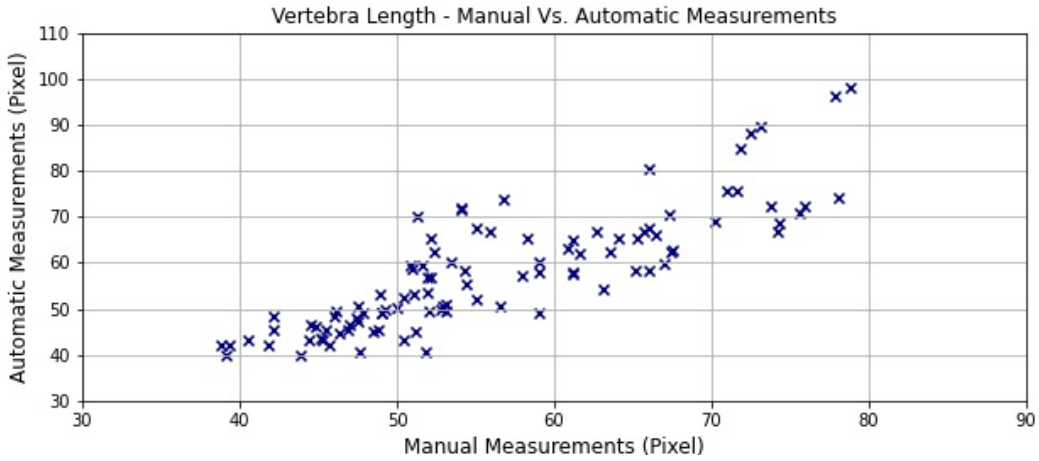


Figure 4.16: Measurement correlation between automated and manual methods of vertebra length quantification.

Pearson Correlation	Mean Absolute Error		p-value
0.85	5.03 pixels	0.05 mm	0.2905

Table 4.5: Summary of automatic vs. manual measurements for vertebra length quantification.

4.5 Summary

The results outlined in this chapter are of high quality and provide an excellent benchmark for the standard of new pipelines that will evolve due to this study. These results will be further analysed in-depth in the next chapter.

CHAPTER

FIVE

DISCUSSION

5.1 Introduction

For each section outlined in this chapter, the results of chapter 4 are discussed in detail with respect to the effects of the methods employed in chapter 3. Following this, the issues that arose throughout this project and subsequent limitations will be addressed along with methods that could of been employed to rectify these problems. The impact of each framework with respect to the pipeline and overall field of the analysis of zebrafish phenotypes and translational genomics will then be discussed. To conclude, the avenues in place for future work are outlined.

5.2 Analysis of Findings

5.2.1 Automatic Segmentation Models

Figure 4.1 and Figure 4.7 reflect the learning curves of the constituent models of both the spine and vertebrae segmentation ensemble models over each epoch during respective training stages. The fit and generalisation of each model to the respective training and hold-out validation dataset respectively can be determined when analysing corresponding learning curves. Generally, a well-fit model that has been trained over an appropriate time or amount of epochs can be visualised in Figure 5.1.

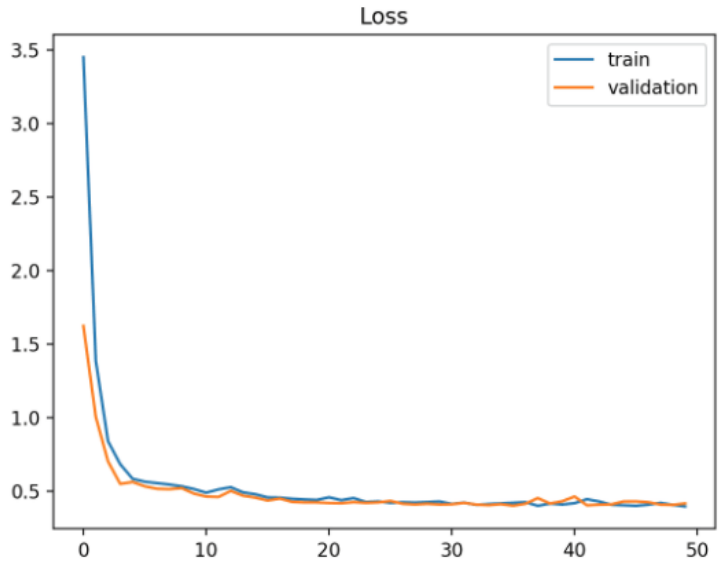


Figure 5.1: Sample schematic of a well-fit and appropriately trained model. Both training and validation loss reduce and converge to the point of stability. Gap between training and validation loss remains minimal.

By analysing the learning curves and relating them to the effects of the implemented hyperparameters of a model, the NOs and its relative performance can be further understood.

Spine Segmentation Models

Analysing the learning curves of the constituent spine segmentation models visualised in Figure 4.1, it can be seen that Model 3's learning performance was the most similar to the reference model depicted in Figure 5.1. Its training and validation curves reduce and converge to a stable point with a minimal gap between each. From this, it was expected that this model would generalize well to an external testing dataset. This is confirmed visually in Figure 4.4 where the raw NOs of the random samples are very clear and well-defined against the significantly darker background. Finally, the performance evaluation in Table 4.1 further confirms this synopsis whereby the model achieves the highest results with respect to Precision, Sensitivity and subsequently F1 Scores. These results are of the same caliber and standard as the work outlined in section 2.3.

Model 4's learning performance was similar to the reference model and Model 3 as both losses generally reduced over epoch iterations, however the gap between the training and validation curve is significantly larger than that of Model 3 throughout the training stage. As the training curve resided below the validation curve throughout the training stage, it can be understood that the model's training dataset was unrepresentative or could not provide sufficient information for well-rounded training. Referring to Table 3.1, the training datasets of Model 3 and 4 mainly differed as a result of data augmentation whereby various ranges of Gaussian blur and gamma contrast were applied. From this, it can be suggested that Model 4 was slightly over-fitted to its training dataset in comparison to Model 3. This model was expected to perform adequately to an external testing dataset, in comparison to Model 3. Reviewing the NOs of Figure 4.5, the spine segmentation contour is well defined against a significantly darker background than that of Model 3 with the exception of the random misclassified pixels in the centre. These occur specifically on the zero-padded boundary the inputted zebrafish X-ray sample. When the NO is binarized, the effect of these misclassified pixels is dampened however it effects the overall performance of the model when the evaluation metrics in Table 4.1 are reviewed. Model 4 achieved the highest Balanced Accuracy Rate with respect to the other

constituent models. Despite these changes, the model still performed to a similar standard as set in human segmentation models, as discussed in section 2.3.

Model 2's learning curves reduced and converged over epoch iterations, however unlike Model 3 and 4, its validation and training curves crossed after the fourth epoch. This reflects that the validation dataset was easier to predict than the training dataset used at the beginning of the training stage. Referring to Table 3.1, the training dataset was smaller with a larger range of Gaussian blur and gamma contrast applied than that of Model 3 and 4. The vast variety in the smaller training dataset made it more difficult for the model to pinpoint necessary features to make accurate predictions. For the final epochs, the model's learning curves decreased and remained within close proximity to each other. From this, it was expected that Model 2 would perform appropriately with an external testing dataset. Reviewing the NOs of Figure 4.3, the spine segmentation contour is well defined and similar to that of Model 3. However, the background pixels are significantly lighter than previous models reviewed. This is a direct result of the variety of gamma contrast used on the smaller sample size. This brighter background will introduce the potential for the occurrence more misclassified pixels in the surrounding region of the true segmentation, further effecting the evaluation metrics, as seen in Table 4.1.

Model 1's learning curve did not correlate with that of Figure 5.1. Similar to that of Model 2, its validation and training curves crossed after the fourth epoch. This was to be expected as the hyperparameters for both models were quite similar. However, following the seventh epoch, the learning curves of Model 1 diverge dramatically. This reflects Model 1 increasingly over-fitting to its original training dataset over its larger number of epochs. Referring to Table 3.1, this model differed from all other models as it had a smaller batch size. This model was expected to perform the worst out of all other constituent spine segmentation models, which is further reflected in the evaluation metrics of Table 4.1. Reviewing the NOs of Figure 4.2, the model is clearly effected by the bright spots on the stomach area of the larger fish which leads to the poorer performance of the model. This being discussed, its result were still very similar to that of the works outlined in section 2.3.

Across the board, the constituent spinal segmentation models produced excellent results which were comparable to the studies outlined in chapter 2 involving humans. As a result of these scores, the final spine segmentation ensemble model was another high achiever with a final Dice Similarity Coefficient of 0.92 as seen in Table 4.1. As tested under a 95% Confidence Interval, this was a statistically significant increase in comparison to the other constituent models. With regards to the other evaluation metrics, the ensemble model remained within their common intervals. This was to be expected as ensemble models tend to prove exceptional when based on a group of models with a large variety of performances which is unseen in this particular section.

All models each analysed and produced 100 high quality spine segmentation masks in approximately 5 minutes. This is immensely quicker than the time taken to manually annotate the spine segmentation GTs. The ensemble will be used over Model 3, regardless of the variations in Precision, Sensitivity and F1 Scores performances. It minimises errors as a result of internal biases reflected in the constituent models and produces the highest DSC out of all the other constituent models. These factors make it more clinically acceptable and reliable for use in Dr. Kague's domain.

Vertebrae Segmentation Models

The learning curves of the constituent vertebrae segmentation models, as depicted in Figure 4.7, reflect similar issues to each other. Losses are computed over many iterations of different gradients and converge to the local extreme. The amount they move along the gradient is controlled by the learning rate parameter. In the case of the constituent vertebrae segmentation models, the learning rate parameters employed were too small and converged on local extremes rather than the preferred global extremes. This is seen in Figure 4.7 as after six epochs the training and validation curves remain stagnant in every model.

As previously seen in Model 1 and 2 of the spine segmentation constituent models, the training curves cross the validation curves indicating that every model had difficulty analysing

the important features within the training dataset during the training stages. Reflecting on Table 3.2, there was too much variety in regards to gamma contrast coefficients and Gaussian blur applied during data augmentation which made the data imperceptible.

As a result of the poor choice in learning rate parameters and high level of variation within the training dataset, the resulting learning progressions of the verterbae segmentation models are poor. These models were expected to exhibit more difficulties and much poorer performances than reflected in the constituent spine segmentation models which is further confirmed in Figures 4.8 to 4.11. Across the board, the NOs of the random samples show a clear understanding of the region in which the 10 vertebrae to be segmented are but the background pixels are just too bright. This increases the probability of misclassified pixels occurring due to the brighter background. This issue along with the small size of the vertebrae negatively effected the evaluation metrics between the GT and BOs of each model, as outlined in Table 4.2.

Despite the results of the constituent vertebrae segmentation models, the final vertebrae segmentation ensemble model produced adequate results for the first ever implementation of an automatic vertebrae segmentation model designed specifically for zebrafish. Each model analysed and produced 100 vertebrae segmentation masks in approximately 5 minutes. This is a much larger improvement in time-efficiency than in conparison to the spine constituent models. Overall, the ensemble model achieved a Balanced Accuracy Rate of 0.90, Dice Similarity Coefficient of 0.79 and Sensitivity of 0.80 as outlined in Table 4.2. When tested under a 95% Confidence Interval, these were statistically significant increases in comparison to the other constituent models. These results are not overly far removed from the work of Al Arif *et al.* and Kuok *et al.* in section 2.3 [26, 27, 28].

5.2.2 Segmentation Analysis Framework

All four fully automatic segmentation analysis frameworks performed with positive correlations with respect to the original methods of phenotypic analysis in place for zebrafish. Manual measurements and analysis were time-consuming and subjective. Employing these automatic frameworks to quantify the spine length, spine curvature, vertebra length and vertebra area allows for a faster and more objective approach to phenotypic analysis. These frameworks were employed on the external testing dataset and within 10 minutes an excel sheet was produced recording the spine length, the location and measure of spine curvature, the vertebra length and vertebra area for every sample. On average, the measurements conducted by all four automatic frameworks took approximately 6 seconds for each sample.

Spine Length Quantification

Originally, the length of the zebrafish spine was quantified using the GUI, ImageJ. Dr. Kague had to overlay smaller line segments to map the spine contour to approximate the length. This was inefficient as only one zebrafish could be analysed at a time making it an extremely time-consuming and labour-intensive task to analyse large quantities of samples during functional screening.

The new automatic framework, outlined in subsection 3.4.1, used to quantify spine length exhibited a very strong positive Pearson Correlation of 0.97 with the manual measurements performed by Dr. Kague, as visualised in Figure 4.13. It was also determined that there was no statistically significant difference between the two methods of measurements. As a result of the exceptional results highlighted in Table 4.3 and the highly efficient spine segmentation ensemble model, further work can be done to connect these two frameworks together to allow for the generation of an exemplary tool to aid in the phenotypic analysis of zebrafish spine length quantification.

Spine Curvature Quantification

As previously highlighted in subsection 3.4.2, spine curvature in zebrafish was analysed based on a qualitative metric using Equation 3.7. From this, the average value, maximum value and standard deviation of the zebrafish spine curvature was analysed and used to compare general curvature between samples based on a common scale. This framework could not provide quantitative metrics or the curvature's position relative to the anterior/posterior ends of the zebrafish (Figure A.1).

The new automatic framework, outlined in chapter 3, used to quantify the spine curvature in zebrafish exhibited a positive Pearson Correlation of 0.64 with that of the previous framework in place, as visualised in Figure 4.14. Analysing this figure visually, it can be seen that the main cohort of samples are strongly correlated apart from the one sample in the top left corner, (-1.6, 0.5). This sample has been visualised in Figure 5.2. As can be seen in the figure, both frameworks were drawn to different curvature points on the spine. The original framework indicated the maximum curvature of this particular sample occurring at the posterior end of the spine, as visualised by the more brightly coloured clusters of points. The new framework identified that particular section as an inflection point within its polynomial (yellow point), subsequently measuring the more anterior curved section of the zebrafish spine.

The new framework is more robust and objective compared to that of the original framework. It provides fast, quantitative measurements that can be used to constitute a meaningful analysis based on one fish rather than relying on a qualitative colour scale as in the original framework. This new framework also indicates the exact position of the extreme points in which it quantifies spine curvature, allowing for a description of relative curvature locations. From this, a platform has been set such that spine curvatures relative to a ends of the zebrafish (anterior/posterior) can be isolated and analysed specifically which can further relate to possible applications of studies involving skeletal conditions causing kyphotic/lordotic spine curvature.

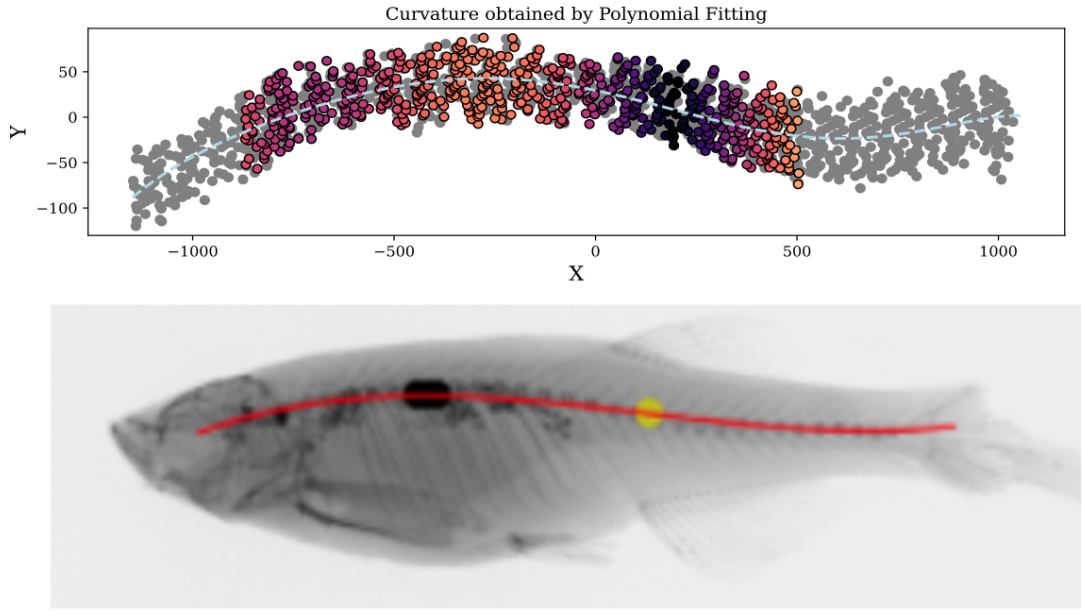


Figure 5.2: Sample case I20200309144541.3. Outlier flagged through the analysis of the spine curvature measurement correlations performed by original and new frameworks. As can be seen, the original framework (top) identified an emphasis on the more posterior curve of the spine, as visualised by the brighter coloured cluster of points. The new framework (bottom) measured the more anterior curve of the spine, as visualised by the black spot, indicating the extreme curvature point.

Vertebra Area Quantification

The vertebra area could originally be quantified by Dr. Kague using the GUI, ImageJ. This was inefficient as each vertebra within the ROI as outlined in section 3.3.1 had to be identified and analysed individually. Relating back to the time it takes to manually annotate the vertebrae segmentations in section 3.3.1, this was time-intensive with regards to the analysis of a single zebrafish without the consideration of the thousands of other samples needed to be analysed during functional screening.

The new automatic framework, outlined in subsection 3.4.3, used to quantify vertebra area exhibited a very strong positive Pearson Correlation of 0.92 with the manual measurements performed by Dr. Kague, as visualised in Figure 4.15. It was also determined that there was no statistically significant difference between the two methods of measurements. The vertebra segmentation ensemble model performed adequately, but not to the standard in which

this framework can be directly linked to its BOs. Until the performances of the ensemble model improve, this framework should be limited to use with manually annotated vertebrae segmentation masks only with regards to the application of phenotypic analysis of zebrafish.

Vertebra Length Quantification

As with spine length and vertebra area, the vertebra length was originally measured by Dr. Kague using the GUI ImageJ. It posed the same efficiency issues as the quantification of vertebra area due to the relative size of each vertebra and sample size with relation to one single zebrafish. Subsequently, this analysis became excessively labour-intensive and time-consuming for one single zebrafish without the consideration of the thousands of other samples needed to be analysed during functional screening.

The new automatic framework, outlined in subsection 3.4.4, used to quantify vertebra area exhibited a very strong positive Pearson Correlation of 0.85 with the manual measurements performed by Dr. Kague, as visualised in Figure 4.16. It was also determined that there was no statistically significant difference between the two methods of measurements. This framework is limited to being an analysis aid for Dr. Kague by using manually annotated vertebrae segmentation masks. Similar to the vertebrae area quantification framework, it cannot be used to analyse the BOs of the vertebrae segmentation ensemble model until it is modified such that its performance improves.

5.3 Limitations

This study was mainly implicated as a result of the size of the data handled and time. At the very beginning of this project, Dr. Kague supplied 200 X-rays with samples of CaHA and six zebrafish in each. Each X-ray DICOM file was 4800 pixels by 6080 pixels subsequently making the average zebrafish X-ray sample approximately 4000 pixels by 1000 pixels after they were separated, as discussed in section 3.2. This resulted in the rectangular samples visualised throughout chapter 3. These samples were then zero-padded into square areas of approximately 4000 pixels by 4000 pixels as outlined in section 2.2.3, following the examples of section 2.3 and also visualised throughout chapter 4.

Training of each segmentation model was carried out on Google Colaboratory Pro using the NVIDIA Tesla K80 GPU. This platform had a time constraint in which one user could have access to the GPU for a maximum of 24 hours. Exceeding this, the platform would disconnect and all work previous would be terminated.

To ensure adequate training of each model was time-efficient and completed within the confines of the time limits imposed by Google Colaboratory Pro, all zero-padded square samples were reduced to tensors of sizes [256, 256]. With regards to the spine segmentation models, this was appropriate as the spine regions were large enough to be visible in both the original rectangular X-ray samples and the smaller [256, 256] tensor samples. This is further validated visually in the learning curves depicted in Figure 4.1, the GT comparisons of Figures 4.8 to 4.11 and performance evaluations of Table 4.1. From this, the spine segmentation ensemble model performs at a high enough level to allow the spine length and spine curvature quantification frameworks to be aligned and form the novel pipeline, fulfilling the first specific objective outlined in section 1.2.

The vertebrae were difficult to be seen in the original rectangular X-ray samples, as visualised in Figure 3.7. When the rectangular samples are reduced to the smaller [256, 256] tensors, they became even more difficult to visualise. This caused pitfalls in training the constituent vertebrae segmentation models as it would be very difficult to isolate the

important features within the inputs to make accurate predictions. This subsequently resulted in the different learning curves of the constituent vertebrae segmentation models in Figure 4.7 compared to the constituent spine segmentation models in Figure 4.1, and the weaker performance evaluation metrics outlined in Table 4.2. This prevents the vertebra area and length quantification frameworks from being directly concatenated to the vertebrae segmentation ensemble model to form the novel pipeline. As a result, the second objective was partially fulfilled whereby a novel and potentially reliable pipeline was designed and the quantification frameworks for vertebra area and length performed to a similar standard as Dr. Kague. However, due to the average performance of the ensemble model, this specific objective falls just short of being fully completed.

Reflecting on this work, this limitation could have been rectified in one of the two possible ways. Firstly, another step which defined the region in which the vertebrae of interest would be should have been generated before the development of the vertebrae segmentation model as outlined by Al Arif *et. al* and Kuok *et. al* in section 2.3. This would of introduced a narrowed window which could of visualised the vertebrae better, further enhancing predictions of the final vertebrae segmentation models. This could have been implemented following the steps visualised in Figure 5.3. If there was more time allocated for this project, this avenue would have been explored.

The size of the data handled could also have been significantly reduced. Following deliberations with members involved in this Zebrafish Project surrounding methods of progressing this work further over the next few months, new emerging techniques of data handling were discussed. This involved training CNNs on rectangular shaped inputs with dimensions still of the order of 2^n, for example [256, 128], instead of the original square shape [45]. By employing this method, the network inputs would have been smaller allowing the vertebrae to be more distinguishable within sample which would further allow some improvement in the overall verterbae segmentation models. The reduction in the size of samples to process within the networks would have also reduced the training times for the models developed in Google Colaboratory Pro.

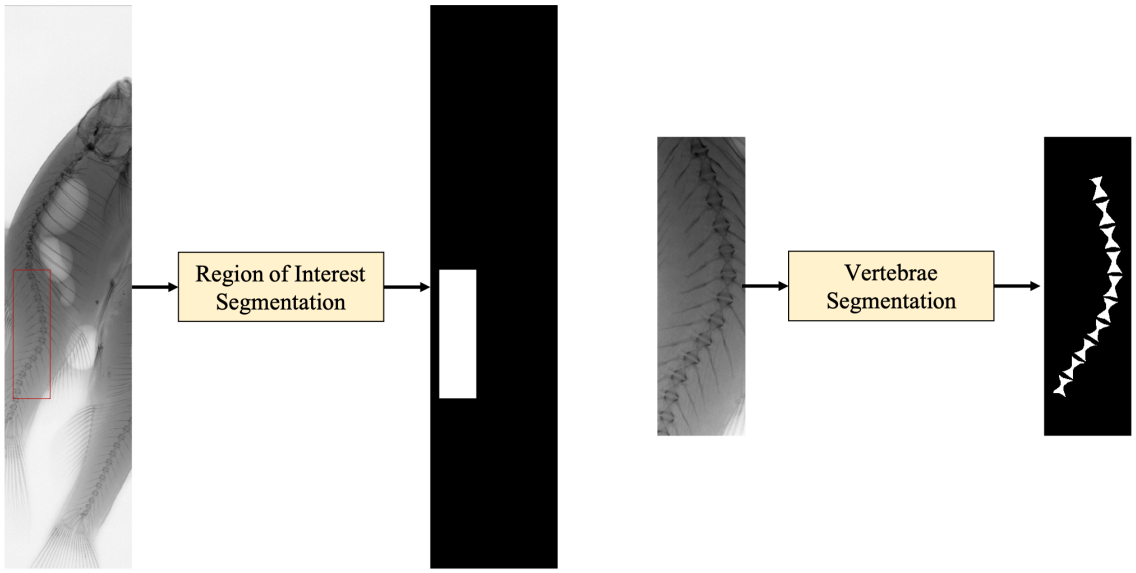


Figure 5.3: Visualisation of the two-step vertebrae segmentation model. The first step involves isolating the region of interest which contains the vertebrae to be segmented using a neural network. This region lies below the ribs and between the anal and dorsal fins. Following this, the network output is merged with the network input. The product is cropped and further inputted into another neural network to identify the vertebrae of interest.

5.4 Impact

Previous work to segment the vertebrae in the entire zebrafish spine to further quantify spine curvature was previously attempted however it fell short as it was only a semi-automatic approach that could be applied to only one zebrafish at a time and extremely image dependent. The design of a pipeline in which the zebrafish vertebral column can be segmented using neural networks and allow further analysis using completely novel automatic quantification frameworks has not been implemented before. There is no other publications that relate the phenotypic screening of skeletal conditions through the combined use of zebrafish and AI. The work outlined in this project is the first ever of its kind.

The spine segmentation pipeline shows huge promise as an aid for Dr. Kague in the analysis of the spine length and curvature. The spine segmentation ensemble model achieved an excellent DSC of 0.92, Precision of 0.96 and Sensitivity of 0.89. The automatic frameworks

developed for the quantification of spine length and curvature performed such that there was no statistically significant difference between their measurements and those performed by a domain expert. These frameworks also performed much quicker than any domain expert could. The new spine curvature quantification framework also evaluates the position of the curvature relative to the anterior and posterior ends of the zebrafish spine. This further opens avenues past the study of identifying causal genes for osteoporosis and into the study of identifying causal genes for kyphotic and lordotic bending of the spine.

The vertebrae segmentation pipeline will be a continuing work in progress. The performance of vertebrae segmentation ensemble model is not to the standard in which it can be relied upon to produce valid vertebrae segmentation models for further appropriate analysis. That being said, its final performance is a solid benchmark to work from and later build upon. The automatic quantification framework for the novel vertebrae analysis pipeline are ready for implementation whenever the performance of the vertebrae segmentation model improves to a sufficient standard.

The work outlined in this project is completely novel and is of extremely high quality in terms of performance evaluations. No other research study involves the development of one neural network to carry out the segmentation of the vertebral column of zebrafish using X-rays, let alone the eight generated through this study and the two final ensemble models. The automatic frameworks defined in this study are also highly innovative while producing excellent quality results similar to that of domain experts. To be specific, the Cobb Angle method has not been adapted in the way it has been outlined in subsection 3.4.2 to further be used on zebrafish. As a result of all the work accomplished in this study, an Invention Disclosure Form (IDF) with University College Dublin has been filed for to document the development of both the spine and vertebrae segmentation and analysis pipelines to protect the intellectual property.

5.5 Future Work

The Zebrafish Team plan to push this study further and broaden its horizons in the following ways:

1. Improve the vertebrae segmentation models to the level at which the spine segmentation models perform. By accomplishing this, all specific objectives outlined in section 1.2 are fully achieved and rapid screening for therapeutic targets of osteoporosis can begin. This will be achieved by employing one method outlined in section 5.3, namely introducing the automatic ROI segmentation step ahead of the final vertebrae segmentation model.

2. The code developed throughout this study will be documented following quality management procedures. FDA and CE approval will later be applied for.

3. The automatic framework will further be used to develop a classification model to discern differences between diseased mutated zebrafish and normal/wildtype zebrafish.

4. The vertebrae segmentation model will be employed for the comparative analysis of bone mineral density within the vertebrae of interest to effectively evaluate potential genetic target for osteoporosis. Yushi Yang had previously developed the framework used to semi-automatically segment the spine regions for the spine segmentation model's GTs as discussed in section 3.3.1 [33]. The steps outlined in Figure 3.6 originally fit a polynomial to the spinal landmarks and was further extrapolated onto the X-ray to analyse the variation in pixel intensities along the vertebra column. This resulted in distributions visualised in Figure 5.4. As can be seen in the figure, there is clear peaks where the vertebrae appear and troughs within the intervertebral discs. By employing a reliable vertebrae segmentation model, vertebra regions should be more distinguishable, troughs should be completely minimised and noise across the distribution should be reduced. The pixel intensity ranges of the vertebra can then be compared appropriately to the pixel intensities of the known CaHA samples to further evaluate the bone mineral densities of the zebrafish sample's vertebrae. By analysing fluctuations of these pixel

intensities - correlating to bone mineral densities - between the mutated zebrafish, genetic targets for osteoporosis can be identified.

5. The vertebrae segmentation model or the subsequent automatic analysis framework will be modified to aid studies investigating intervertebral disc diseases in zebrafish. A sample depicting intervertebral disc disease is visualised in Figure 5.5. As outlined in subsection 3.4.4, a 2nd order polynomial was fit to the vertebrae segmentation mask to analyse the vertebra length. By inverting the outputs of the vertebrae segmentation model and fitting the same polynomial to the X-ray sample, the pixel intensities of the intervertebral discs of mutated zebrafish can be analysed and compared to determine causal genetic targets, similar to the processes visualised in Figure 5.4.

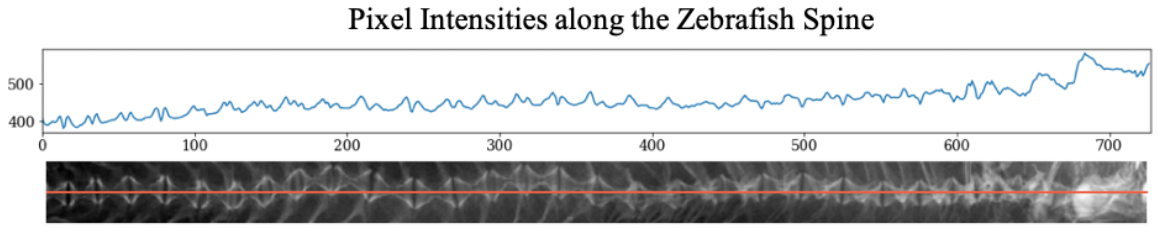
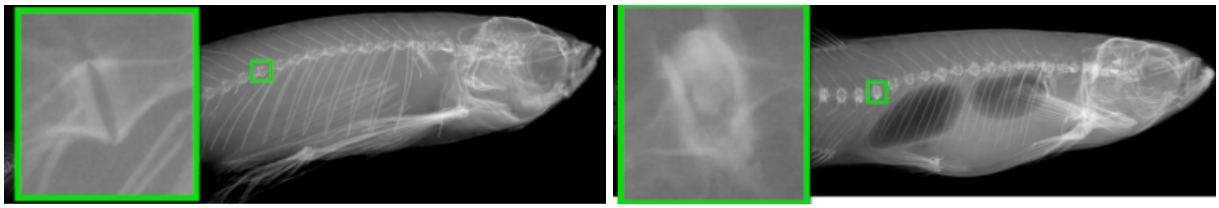


Figure 5.4: Visualisation of the variation in pixel intensities along the spine of the zebrafish. The figure highlights the peak intensities relating to the verterbae and troughs relating to the intervertebral discs. Depicted is also the steady rise in pixel intensities as the sample is investigated closer to the anterior end of the spine. This rise is due to the interference of the ribs.



(a) Healthy Intervertebral Disc Sample **(b)** Diseased Intervertebral Disc Sample

Figure 5.5: Visual comparison between healthy and diseased intervertebral discs. Healthy intervertebral discs have a clear/empty space between the vertebrae. Diseased intervertebral discs have reminant spots between the vertebrae. These can be identified by comparing ranges of pixel intensities between normal and abnormal samples.

CHAPTER

SIX

CONCLUSION

The work outlined in this project is completely novel and is of extremely high quality in terms of performance evaluations and innovative design. Two pipelines for the segmentation and subsequent analysis of the zebrafish spine and vertebrae were developed for the first time. This involved the development of eight segmentation models converging into two different ensemble models tailored specifically for the zebrafish alongside the generation of four new, fully automatic frameworks designed for fast and robust analysis of the vertebral column with subsequent performances similar to that of domain experts. Publications of the caliber achieved by this work are similar to the automatic segmentation and analysis of the vertebral column in humans. Previous work in this field collaborating zebrafish and AI has been limited to semi-automatic approaches of vertebrae segmentation. This study will provide a steady base platform for potential enhancements and improvements to come with regards to projects mixing fish models with AI.

The tasks exceptionally performed in this study will promote rapid genetic screening and reliable analysis of zebrafish phenotypes resulting from genetic mutations. The code developed will be documented and protected using an Invention Disclosure Form, following quality management procedures and applications for FDA and CE approval will be sought for. These

perocedures will allow the work done here to be shared and used internationally by the genetic community for different genetic screening processes. The analysis frameworks outlined in this study can be further used for a classification model to discern between diseased mutant zebrafish fish and normal/wild zebrafish. Future work will capitalise on the proven work carried out in this project by adapting the framework to the identification of therapeutic targets of osteoporosis and intervertebral disc diseases.

Bibliography

BIBLIOGRAPHY

[1] Francesca Tonelli, Jan Willem Bek, Roberta Besio, Adelbert De Clercq, Laura Leoni, Phil Salmon, Paul J. Coucke, Andy Willaert, and Antonella Forlino. Zebrafish: A Resourceful Vertebrate Model to Investigate Skeletal Disorders. *Frontiers in Endocrinology*, 11(July), 2020.

[2] Ivan Dieb Miziara, Ana Tereza de Matos Magalhães, Maruska d'Aparecida Santos, Érika Ferreira Gomes, and Reinaldo Ayer de Oliveira. Research ethics in animal models. *Brazilian Journal of Otorhinolaryngology*, 78(2):128–131, 2012.

[3] Marta Carnovali, Giuseppe Banfi, and Massimo Mariotti. Zebrafish Models of Human Skeletal Disorders: Embryo and Adult Swimming Together. *BioMed Research International*, 2019, 2019.

[4] Robert Bryson-Richardson, Silke Berger, and Peter Currie. Introduction. *Atlas of Zebrafish Development*, pages 1–2, 2012.

[5] Dylan J.M. Bergen, Erika Kague, and Chrissy L. Hammond. Zebrafish as an emerging model for osteoporosis: A primary testing platform for screening new osteo-active compounds. *Frontiers in Endocrinology*, 10(JAN):1–20, 2019.

[6] Canyon Hydro, Executive Summary, Finding Of, T H E Potential, Xavier Reales Ferreres, Albert Raurell Font, Azliza Ibrahim, Niyonsaba Maximilien, D Lumbroso, A Hurford,

J Winpenny, S Wade, Robert T Sataloff, Michael M Johns, Karen M Kost, The State-of-the art, The Motivation, 2 Norsuzila Ya'acob1, Mardina Abdullah1, 2 and Mahamod Ismail1, M. Medina, T. L. Talarico, I. A. Casas, T. C. Chung, W. J. Dobrogosz, L. Axelsson, S. E. Lindgren, W. J. Dobrogosz, Leila Kerkeni, Paula Ruano, Lismet Lazo Delgado, Sergio Picco, Liliana Villegas, Franco Tonelli, Mario Merlo, Javier Rigau, Dario Diaz, and Martin Masuelli. We are IntechOpen , the world ' s leading publisher of Open Access books Built by scientists , for scientists TOP 1 %. *Intech*, 32(July):137–144, 2013.

[7] Ralf Dahm and Robert Geisler. Learning from small fry: the zebrafish as a genetic model organism for aquaculture fish species. *Marine biotechnology*, 8(4):329–345, 2006.

[8] Andreas Holzinger, Andre Carrington, and Heimo Muller. Measuring the quality of explanations: The System Causability Scale (SCS), 2019.

[9] Joseph E. Burns, Jianhua Yao, and Ronald M. Summers. Artificial Intelligence in Musculoskeletal Imaging: A Paradigm Shift. *Journal of Bone and Mineral Research*, 35(1):28–35, 2020.

[10] Nahian Siddique, Paheding Sidike, Colin Elkin, and Vijay Devabhaktuni. U-Net and its variants for medical image segmentation: theory and applications. 2020.

[11] Dennis Segebarth, Matthias Griebel, Nikolai Stein, Cora R. von Collenberg, Corinna Martin, Dominik Fiedler, Lucas B. Comeras, Anupam Sah, Victoria Schoeffler, Teresa Lüffe, Alexander Dürr, Rohini Gupta, Manju Sasi, Christina Lillesaar, Maren D. Lange, Ramon O. Tasan, Nicolas Singewald, Hans Christian Pape, Christoph M. Flath, and Robert Blum. On the objectivity, reliability, and validity of deep learning enabled bioimage analyses. *eLife*, 9:1–36, 2020.

[12] Cosmin Cernazanu-glavan and Stefan Holban. 2013] Segmentation of bone in X-ray images using CNN. *Advances in Electrical and Computer Engineering*, 13(1)(x), 2013.

[13] Van Hiep Phung and Eun Joo Rhee. A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets. *Applied Sciences*, 9(21), 2019.

[14] Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi. Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015: 18th International Conference Munich, Germany, October 5-9, 2015 proceedings, part III. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9351(Cvd):12–20, 2015.

[15] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[16] Kyoung-Su Oh and Keechul Jung. GPU implementation of neural networks. *Pattern Recognition*, 37(6):1311–1314, 2004.

[17] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020.

[18] Yunyan Zhang, Daphne Hong, Daniel McClement, Olayinka Oladosu, Glen Pridham, and Garth Slaney. Grad-CAM helps interpret the deep learning models trained to classify multiple sclerosis types using clinical brain magnetic resonance imaging. *Journal of Neuroscience Methods*, 353:109098, 2021.

[19] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[20] Cristina Stolojescu-Crişan and Ştefan Holban. A Comparison of X-ray image segmentation techniques. *Advances in Electrical and Computer Engineering*, 13(3):85–92, 2013.

[21] Liansheng Wang, Qiuhao Xu, Stephanie Leung, Jonathan Chung, Bo Chen, and Shuo Li. Accurate automated Cobb angles estimation using multi-view extrapolation net. *Medical Image Analysis*, 58:101542, 2019.

[22] Hongbo Wu, Chris Bailey, Parham Rasoulinejad, and Shuo Li. Automated comprehensive Adolescent Idiopathic Scoliosis assessment using MVC-Net. *Medical Image Analysis*, 48:1–11, 2018.

[23] Yongcheng Tu, Nian Wang, Fei Tong, and Hemu Chen. Automatic measurement algorithm of scoliosis Cobb angle based on deep learning. *Journal of Physics: Conference Series*, 1187(4), 2019.

[24] Alan Petrônio Pinheiro, Júlio Cézar Coelho, Antônio C Paschoarelli Veiga, and Tomaž Vrtovec. A computerized method for evaluating scoliotic deformities using elliptical pattern recognition in X-ray spine images. *Computer methods and programs in biomedicine*, 161:85–92, 2018.

[25] Omar Al Okashi, Hongbo Du, and Hisham Al-Assam. Automatic spine curvature estimation from X-ray images of a mouse model. *Computer Methods and Programs in Biomedicine*, 140:175–184, 2017.

[26] Chan-Pang Kuok, Min-Jun Fu, Chii-Jen Lin, Ming-Huwi Horng, and Yung-Nien Sun. Vertebrae Segmentation from X-ray Images Using Convolutional Neural Network. In *Proceedings of the 2018 International Conference on Information Hiding and Image Processing*, pages 57–61, 2018.

[27] S. M. Masudur Rahman Al Arif, Karen Knapp, and Greg Slabaugh. Fully automatic cervical vertebrae segmentation framework for X-ray images. *Computer Methods and Programs in Biomedicine*, 157:95–111, 2018.

[28] SM Masudur Rahman Al Arif, Karen Knapp, and Greg Slabaugh. Shape-aware deep convolutional neural network for vertebrae segmentation. In *International Workshop on Computational Methods and Clinical Applications in Musculoskeletal Imaging*, pages 12–24. Springer, 2017.

[29] Shannon Allen, Eric Parent, Maziyar Khorasani, Douglas L Hill, Edmond Lou, and James V Raso. Validity and reliability of active shape models for the estimation of Cobb angle in patients with adolescent idiopathic scoliosis. *Journal of digital imaging*, 21(2):208–218, 2008.

[30] Junhua Zhang, Edmond Lou, Lawrence H Le, Douglas L Hill, James V Raso, and Yuanyuan Wang. Automatic Cobb measurement of scoliosis based on fuzzy Hough transform with vertebral shape prior. *Journal of digital imaging*, 22(5):463, 2009.

[31] Lucy M McGowan, Erika Kague, Alistair Vorster, Elis Newham, Stephen Cross, and Chrissy L Hammond. Wnt16 Elicits a Protective Effect Against Fractures and Supports Bone Repair in Zebrafish. *JBMR plus*, 5(3):e10461, 2021.

[32] Paco López-Cuevas, Luke Deane, Yushi Yang, Chrissy L Hammond, and Erika Kague. Transformed notochordal cells trigger chronic wounds in zebrafish, destabilizing the vertebral column and bone homeostasis. *Disease models & mechanisms*, 14(3), 2021.

[33] Yushi Yang. zefiia. `https://github.com/yangyushi/zefiia.git`, 2020.

[34] Johannes Schmidt. PyTorch-2D-3D-UNet-Tutorial. `https://github.com/johschmidt42/PyTorch-2D-3D-UNet-Tutorial/blob/master/unet.py`, 2020.

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach,

H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[36] Ekaba Bisong. *Google Colaboratory*, pages 59–64. Apress, Berkeley, CA, 2019.

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[38] Silvia Sapora, Bogdan Lazarescu, and Christo Lolov. Absit invidia verbo: Comparing deep learning methods for offensive language. *arXiv preprint arXiv:1903.05929*, 2019.

[39] Sofia Visa, Brian Ramsay, Anca L Ralescu, and Esther Van Der Knaap. Confusion Matrix-based Feature Selection. *MAICS*, 710:120–127, 2011.

[40] Norio Yamamoto, Shintaro Sukegawa, Akira Kitamura, Ryosuke Goto, Tomoyuki Noda, Keisuke Nakano, Kiyofumi Takabatake, Hotaka Kawai, Hitoshi Nagatsuka, Keisuke Kawasaki, et al. Deep Learning for Osteoporosis Classification Using Hip Radiographs and Patient Clinical Covariates. *Biomolecules*, 10(11):1534, 2020.

[41] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: Image processing in python. *PeerJ*, 2:e453, 2014.

[42] Aristidis Likas, Nikos Vlassis, and Jakob J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36(2):451–461, 2003. Biometrics.

[43] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[44] C. T. Rueden, J. Schindelin, and M. C. Hiner. ImageJ2: ImageJ for the next generation of scientific image data, 2017.

[45] Qin Zou, Hanwen Jiang, Qiyu Dai, Yuanhao Yue, Long Chen, and Qian Wang. Robust Lane Detection From Continuous Driving Scenes Using Deep Neural Networks. *IEEE Transactions on Vehicular Technology*, 69(1):41–54, 2020.

[46] Lumen Learning. Biology for Majors II: Vertebrate Axis Formation.

[47] Nathan C Bird and Paula M Mabee. Developmental morphology of the axial skeleton of the zebrafish, danio rerio (ostariophysi: Cyprinidae). *Developmental dynamics: an official publication of the American Association of Anatomists*, 228(3):337–357, 2003.

Appendices

APPENDIX

A

ANATOMY OF ZEBRAFISH

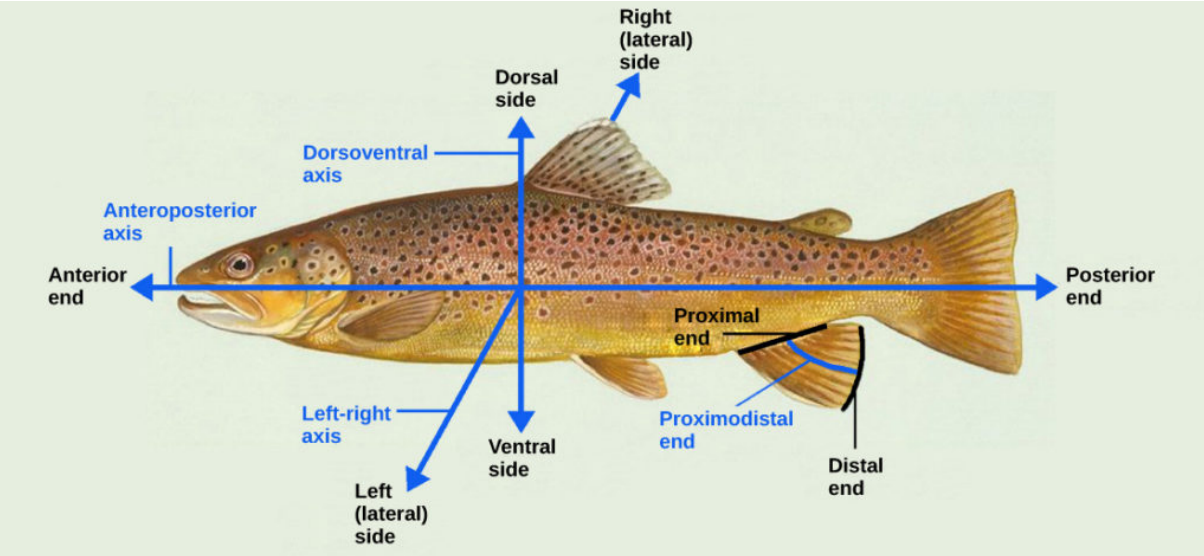


Figure A.1: Reference diagram relating to the anatomical axes and planes of a fish's anatomy, provided by Dr. Erika Kague, adapted from [**?**].

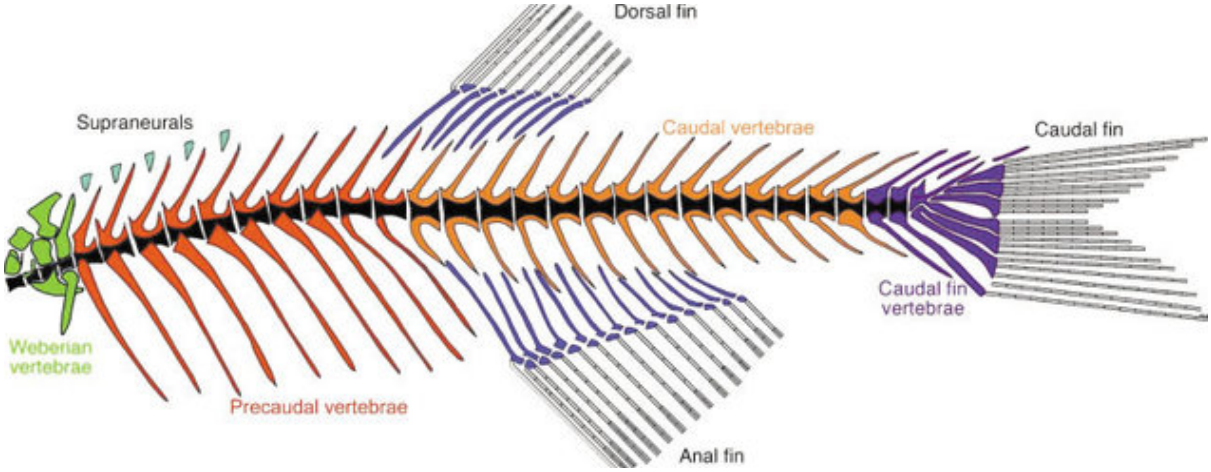


Figure A.2: Generalised diagram of the axial skeleton of a zebrafish, provided by Dr. Erika Kague, adapted from [**?**].

APPENDIX

B

SUMMARY OF U-NET SEGMENTATION MODEL LAYERS

Layer	Type	Output Shape	Number of Parameters
1	2D Convolution	[-1, 32, 256, 256]	320
2	ReLU	[-1, 32, 256, 256]	0
3	2D Batch Normalisation	[-1, 32, 256, 256]	64
4	2D Convolution	[-1, 32, 256, 256]	9248
5	ReLU	[-1, 32, 256, 256]	0
6	2D Batch Normalisation	[-1, 32, 256, 256]	64
7	2D Max-Pooling	[-1, 32, 128, 128]	0
8	Downblock	[-1, 32, 128, 128], [-1, 32, 256, 256]	0
9	2D Convolution	[-1, 64, 128, 128]	18496
10	ReLU	[-1, 64, 128, 128]	0
11	2D Batch Normalisation	[-1, 64, 128, 128]	128
12	2D Convolution	[-1, 64, 128, 128]	36928

13	ReLU	[-1, 64, 128, 128]	0
14	2D Batch Normalisation	[-1, 64, 128, 128]	128
15	2D Max-Pooling	[-1, 64, 128, 128]	0
16	Downblock	[-1, 64, 64, 64], [-1, 64, 128, 128]	0
17	2D Convolution	[-1, 128, 64, 64]	73856
18	ReLU	[-1, 128, 64, 64]	0
19	2D Batch Normalisation	[-1, 128, 64, 64]	256
20	2D Convolution	[-1, 128, 64, 64]	147584
21	ReLU	[-1, 128, 64, 64]	0
22	2D Batch Normalisation	[-1, 128, 64, 64]	256
23	2D Max-Pooling	[-1, 128, 32, 32]	0
24	Downblock	[-1, 128, 32, 32], [-1, 128, 64, 64]	0
25	2D Convolution	[-1, 256, 32, 32]	295168
26	ReLU	[-1, 256, 32, 32]	0
27	2D Batch Normalisation	[-1, 256, 32, 32]	512
28	2D Convolution	[-1, 256, 32, 32]	590080
29	ReLU	[-1, 256, 32, 32]	0
30	2D Batch Normalisation	[-1, 256, 32, 32]	512
31	Downblock	[-1, 256, 32, 32], [-1, 256, 32, 32]	0
32	2D Transposed Convolution	[-1, 128, 64, 64]	131200
33	ReLU	[-1, 128, 64, 64]	0
34	2D Batch Normalisation	[-1, 128, 64, 64]	256
35	Concatenate	[-1, 256, 64, 64]	0
36	2D Convolution	[-1, 128, 64, 64]	295040

37	ReLU	[-1, 128, 64, 64]	0
38	2D Batch Normalisation	[-1, 128, 64, 64]	256
39	2D Convolution	[-1, 128, 64, 64]	147584
40	ReLU	[-1, 128, 64, 64]	0
41	2D Batch Normalisation	[-1, 128, 64, 64]	256
42	Upblock	[-1, 128, 64, 64]	0
43	2D Transposed Convolution	[-1, 64, 128, 128]	32832
44	ReLU	[-1, 64, 128, 128]	0
45	2D Batch Normalisation	[-1, 64, 128, 128]	128
46	Concatenate	[-1, 128, 128, 128]	0
47	2D Convolution	[-1, 64, 128, 128]	73792
48	ReLU	[-1, 64, 128, 128]	0
49	2D Batch Normalisation	[-1, 64, 128, 128]	128
50	2D Convolution	[-1, 64, 128, 128]	36928
51	ReLU	[-1, 64, 128, 128]	0
52	2D Batch Normalisation	[-1, 64, 128, 128]	128
53	Upblock	[-1, 64, 128, 128]	0
54	2D Transposed Convolution	[-1, 32, 256, 256]	8224
55	ReLU	[-1, 32, 256, 256]	0
56	2D Batch Normalisation	[-1, 32, 256, 256]	64
57	Concatenate	[-1, 64, 256, 256]	0
58	2D Convolution	[-1, 32, 256, 256]	18464
59	ReLU	[-1, 32, 256, 256]	0
60	2D Batch Normalisation	[-1, 32, 256, 256]	64
61	2D Convolution	[-1, 32, 256, 256]	9248
62	ReLU	[-1, 32, 256, 256]	0
63	2D Batch Normalisation	[-1, 32, 256, 256]	64

64	Upblock	[-1, 32, 256, 256]	0
65	2D Convolution	[-1, 1, 256, 256]	33

Table B.1: U-Net layer summary provided by PyTorch. Model consists of a total of 1,928,289 trainable parameters.