AN AUTOMATED PIPELINE FOR CONSTRUCTING 3D MODELS OF MONOGENEAN HARDPARTS USING MACHINE LEARNING TECHNIQUE FOR LANDMARKS DETECTION

TEO BEE GUAN

FACULTY OF SCIENCE
UNIVERSITY OF MALAYA
KUALA LUMPUR

2019

AN AUTOMATED PIPELINE FOR CONSTRUCTING 3D MODELS OF MONOGENEAN HARDPARTS USING MACHINE LEARNING TECHNIQUE FOR LANDMARKS DETECTION

TEO BEE GUAN

THESIS SUBMITTED IN FULFILMENT OF THE
REQUIREMENTS FOR THE DEGREE OF DOCTOR OF
PHILOSOPHY

INSTITUTE OF BIOLOGICAL SCIENCES
FACULTY OF SCIENCE
UNIVERSITY OF MALAYA
KUALA LUMPUR

2019

UNIVERSITY OF MALAYA
ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: **TEO BEE GUAN**

Matric No: **SHC110085**

Name of Degree: **DOCTOR OF PHILOSOPHY**

Title of Thesis:

AN AUTOMATED PIPELINE FOR CONSTRUCTING 3D MODELS OF MONOGENEAN HARDPARTS USING MACHINE LEARNING TECHNIQUE FOR LANDMARKS DETECTION

Field of Study:

BIOINFORMATICS

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:

Subscribed and solemnly declared before,

Witness's Signature Date:

Name:

Designation:

AN AUTOMATED PIPELINE FOR CONSTRUCTING 3D MODELS OF MONOGENEAN HARDPARTS USING MACHINE LEARNING TECHNIQUE FOR LANDMARKS DETECTION

ABSTRACT

Organisms in particular parasitic micro-organisms such as monogeneans defy live investigations due to lack of sophisticated equipment and also due to the difficulties in keeping these delicate organisms alive long enough for investigations to be done. The way monogeneans use their various morphological structures to assist them in their survival have not been studied intensively mainly due to the difficulties in manipulating 2D images and illustrations of attaching structures. Laboratory experiments can only provide a snapshot of a system at any given time but do not fully capture the temporal aspect of the system. Researchers have been looking at alternative methods to assist them in their investigation on how monogeneans function. 3D models may aid researchers in studying morphology and function by rotating the 3D models in 360 degrees for 3D visualization. This task is impossible in 2D illustrations or 2D images. However, the development of 3D models is a tedious procedure as one will have to repeat entire complicated modelling process for every new target 3D shape using a comprehensive 3D modelling software. Hence, this study was designed to develop an alternative 3D modelling approach to build 3D models of monogenean anchors which can be used to understand these morphological structures in three dimensions. The aim of this alternative 3D modelling approach is to avoid repeating the tedious modeling procedure for every single target 3D model from scratch. To achieve this aim, an automated 3D modelling pipeline empowered by an Artificial Neural Network (ANN) was developed. This automated 3D modelling pipeline enables automated deformation of a generic 3D model of monogenean anchor into another target 3D anchor. The entire modelling pipeline can be completed with minimum human intervention by only requiring a 2D illustration of target anchor as an input. The ANN was trained by 5000 synthetic 2D illustrations of monogenean anchors using a data augmentation algorithm. Besides, a NoSQL-based database, MongoDB was developed to store all the synthetic 2D illustrations. The stored illustrations datasets were then retrieved to train the ANN. The proposed 3D modelling pipeline empowered by ANN has managed to automate the generation of the 12 target 3D models (representing 12 species: *Cichlidogyrus dracolemma*, *Dactylogyrus wunderi*, *Cichlidogyrus raeymaekersi, Cichlidogyrus aspiralis, Dactylogyrus primaries, Pellucidhaptor merus, Dactylogyrus falcatus, Dactylogyrus vastator, Dactylogyrus pterocleidus, Dactylogyrus falciunguis, Chauhanellus auriculatum* and *Chauhanellus caelatus*) of monogenean anchor from the respective 2D illustrations input without repeating the tedious modeling procedure.

Keywords: Monogenean, landmarks detection, machine learning, noSQL database, 3D modelling.

SATU ALIRAN PEMBINAAN 3D MODEL MORFOLOGI STRUKTUR MONOGENEAN SECARA AUTOMATIK DENGAN MENGGUNAKAN PELAJARAN MESIN TEKNIK UNTUK PENGESANAN MERCU TANDA

ABSTRAK

Organisma terutamanya mikro-organisma seperti monogeneans mengalami kesulitan siasatan langsung kerana kekurangan peralatan canggih dan juga disebabkan oleh kesukaran untuk memastikan organisma itu hidup cukup lama untuk penyiasatan dilakukan. Cara bagaimana monogeneans menggunakan pelbagai struktur morfologi untuk membantu mereka dalam kehidupan mereka tidak dikaji secara intensif disebabkan oleh kesukaran untuk memanipulasi 2D imej dan 2D ilustrasi struktur. Eksperimen makmal hanya dapat memberikan gambaran pada suatu masa tertentu tetapi tidak mampu menangkap aspek temporal sepenuhnya. Penyelidik telah mengkaji kaedah alternatif untuk membantu mereka dalam penyiasatan tentang bagaimana monogenean berfungsi. 3D model boleh membantu para penyelidik mempelajari morfologi dan fungsi mikro organisma melalui putaran 3D model dalam 360 darjah untuk 3D visualisasi. Tugas ini adalah tidak mustahil dalam 2D ilustrasi atau 2D imej. Walau bagaimanapun, pembinaan 3D model adalah prosedur yang sukar kerana seseorang perlu mengulangi keseluruhan proses pemodelan rumit untuk setiap bentuk 3D sasaran baru dengan menggunakan perisian pemodelan 3D yang komprehensif. Oleh itu, kajian ini direka bentuk untuk membangunkan pendekatan pemodelan 3D alternatif untuk membina model 3D penunjuk monogenean yang boleh digunakan untuk memahami struktur morfologi ini dalam tiga dimensi. Pendekatan pemodelan 3D alternatif ini bertujuan untuk mengelakkan mengulang prosedur pemodelan yang rumik bagi setiap model 3D sasaran dari awal. Untuk mencapai matlamat ini, saluran pemodelan 3D automatik yang disokong oleh Rangkaian Neural Buatan (ANN) telah dibangunkan. Pemodelan model pemodelan 3D automatik ini membolehkan satu generic 3D model monogenean berubah bentuk ke sasaran yang lain secara automatik. Seluruh saluran pemodelan dapat diselesaikan dengan mudah dan hanya memerlukan 2D ilustrasi dari sasaran utama sebagai input. ANN dilatih oleh 5000 2D sintetik ilustrasi dengan menggunakan algoritma yang direka untuk penyelidikan ini. Selain itu, pangkalan data berasaskan NoSQL, MongoDB telah dibangunkan untuk menyimpan semua 2D sintetik illustrasi. Data-data ilustrasi yang disimpan kemudiannya diambil untuk melatih ANN. Saluran pembinaan 3D model yang disokong oleh ANN telah berjaya mengotomatikkan generasi 12 3D model (mewakili 12 *spesis Cichlidogyrus dracolemma, Dactylogyrus wunderi, Cichlidogyrus raeymaekersi, Cichlidogyrus aspiralis, Dactylogyrus primaries, Pellucidhaptor merus, Dactylogyrus falcatus, Dactylogyrus vastator, Dactylogyrus pterocleidus, Dactylogyrus falciunguis, Chauhanellus auriculatum dan Chauhanellus caelatus*) daripada anchor monogene dari 2D ilustrasi masing-masing tanpa mengulangi prosedur pemodelan yang rumik.

Kata kunci: Monogenean, pengesanan mercu tanda, pembelajaran mesin, noSQL pangkalan data, pembinaan 3D model.

ACKNOWLEDGEMENTS

Firstly, I would like to express my deepest gratitude to my supervisor, Associate Prof. Dr. Sarinder Kaur, upon her guidance on this project development and thesis writing. Her precious comments on my thesis had helped me a lot in structuring my content so that I manage to explicitly present my work through writing. She has also provided me with a conducive research environment where I could fully utilize all the available equipment to complete my research works. I would also like to express my memorial gratitude to my late supervisor, Prof. Dr. Susan Lim Lee Hong who conceived the initial research idea of this study. Encouraged by her vision about this project, I was motivated to accomplish the objectives and even anticipate taking further step to continue the project development after PhD study. Besides, I am also grateful to Dr Tan Wooi Boon for his generous help in verifying my research outcome based on his expertise in parasitology. On other hand, I would also thank to University of Malaya for offering me studentship and scholarship. Without the sponsorship, I won't be able to focus on my project. Last but not least, I wish to express my deepest love and gratitude to my family who always be my encouragement and support to complete my PhD study regardless of all the hardships I experienced. Thank you all who leave a monumental impact through this entire journey. My heartfelt thanks to you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

ANN : Artificial Neural Network

API : Application Programming Interface

CNN : Convolutional Neural Network

DNN : Deep Neural Network

HTML : Hypertext Markup Language

LTU : Linear Threshold Unit

MLP : Multi-Layer Perceptron

POI : Point of Interest

RNN : Recurrent Neural Network

SVM : Support Vector Machine

3D : Three dimensional

2D : Two dimensional

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

1.1 Background

In taxonomy, it is a common practice to illustrate various anatomical parts (hard parts and soft parts) of an organism, such as a parasite, for species identification and diagnostic purposes. While all organisms are three-dimensional (3D) in nature, the illustrations are two-dimensional (2D) visual representation constrained to selected perspective projection or cutting plane of a 3D structure and therefore provide an incomplete visualization of their structural morphology (Ruthensteiner & Heb, 2008). A digital 3D model, on the other hand, is a 3D representation of the surface of a target object that can give a 3D visual impact to aid researchers in studying the structural and functional morphology of an organism (Laforsch et al., 2012). Over the last decade, the usage of digital 3D model to represent 3D surface of biological specimen has become increasingly common. A digital 3D model enables interactive manipulation of a 3D morphological structure using an intuitive and a simple 3D viewer program (Bates et al., 2009; Berry & Ibbotson, 2010). A digital 3D model can also be easily shared among the research community by hosting it on an online platform such as Sketchfab (Scopigno et al., 2017). Recently, Erolin et al. (2017) presented their work on digitizing specimens from the D'Arcy Thompson Zoology Museum by using micro-CT scanning to produce digital 3D specimens. In comparison with the physical specimens kept in the museum, these digital 3D specimens are available online which can offer wider access to the public to raise awareness about the collection of specimens. Besides, digital 3D models are also used as educational tools to offer students a more engaging learning experience via visualizing and manipulating a 3D biological structure (Kamsin et al., 2007; Lee & Tsai, 2013; Petersson et al., 2009).

Generally, digital 3D models of the biological specimens are developed via a 3D reconstruction process through stacking of serial section images. These serial section images are obtained from 3D imaging modalities such as transmission electron microscope (TEM) (McEwen & Marko, 1999), confocal microscope (Galli et al., 2006; Nair et al., 2017), magnetic resonance microscopy scanner (MRM scanner) (Lee et al., 2015) and X-ray micro-tomography scanner (micro-CT scanner) (Kerbl et al., 2013). These serial section images are then post-processed by a scientific 3D visualization software such as FEI Amira 3D, Able 3D Doctor and Bitplane Imaris, to name a few, to stack them into a digital 3D model. While these 3D imaging modalities along with their associated scientific 3D visualization software are shown to be useful to produce a high-quality digital 3D model, they are usually costly and are not readily available to all researchers. Some of these imaging modalities (e.g. electron tomography) also demand special training to manage tedious specimen preparation prior to usage (Kremer et al., 2015).

Instead of depending on the expensive 3D imaging modalities as aforementioned, Teo et al. (2010) have demonstrated a polygonal modelling method by using 2D illustrations as the templates to construct 3D models of haptoral parts of monogeneans via an existing commercial 3D modelling software, Autodesk 3ds Max. The Autodesk 3ds Max is a widely used 3D software package in the entertainment and architecture industries. The resulting models produced from Autodesk 3ds Max are shown to be effective in offering 3D visualization of the spatial relationship of morphological characters in a monogenean haptor (Teo et al., 2010). However, popular 3D software packages such as Autodesk 3ds Max, Autodesk Maya and LightWave demand great skill to create a decent 3D shape of a target specimen due to the high complexity of the software user interface (Huang et al., 2017). One will need to repeat the complex polygonal modelling steps to create each 3D shape from scratch. Despite the limitations as mentioned above, the modelling work presented by Teo et al. (2010) still shows the possibility to develop a digital 3D model from a 2D illustration without relying on the expensive 3D imaging modalities.

Apart from the above-mentioned methods, it is also possible to construct a new digital 3D model from an existing 3D model through a shape deformation technique (Murakawa et al., 2006). The core idea of this shape deformation technique is to develop a digital generic 3D model which can be deformed into a new 3D shape by aligning the vertices of the digital generic model to match with their corresponding landmark points assigned on an input 2D image. The vertices are a sequence of point primitives used to form the underlying structure of a digital 3D model whereas the landmark points are a set of salient feature points assigned on some selected corners or tips of an input image. The movement of the vertices to match with their corresponding landmark points on an input 2D image will deform the digital generic 3D model and result in a new 3D shape. This shape deformation technique obviates the need of repeating a complicated modelling process to construct every new 3D model from scratch. However, this modelling approach demands a modeller to manually choose the vertices of a digital generic 3D model and align them with the landmark points of an input 2D image. If the alignment between the vertices of a digital generic 3D model with the landmark points of an input 2D image can be automated with minimal human intervention, the entire modelling procedure may become much easier and efficient.

A landmark point detection empowered by a machine learning algorithm could possibly be a solution to enable auto-detection of the landmark points on an input image. Subsequently, the landmark points predicted by the machine learning model can be used to direct the deformation of a digital generic 3D model by automatically mapping its vertices with their corresponding landmark points and eventually produce a target 3D shape. Active research on using machine learning algorithms for landmark point detection has been found on several fields of studies such as human face recognition (Cevikalp et al., 2013; Ranjan et al., 2017), geometric morphometric for bioimages (Vandaele et al., 2018), anatomical features detection for Fast Alzheimer's disease diagnosis (Zhang et al.,

2016) and human body pose detection (Jang & Lee, 2017). To the best of our knowledge, there are no published records on application of machine learning algorithms for landmark point detection to enable automated construction of the digital 3D models for biological specimens such as monogeneans, which is the target specimen of this study.

It is necessary here to briefly describe the monogeneans and the rationale for developing the digital 3D models for the morphological characteristic parts of this microorganism. Monogeneans (Class: Monogenea, Phylum: Platyhelminthes) are parasitic flatworms which can be divided into 3 subclasses, viz. Polystomatoinea Lebedev, 1986, Oligochoinea Bychowsky, 1937 and Polyonchoinea Bychowsky, 1937 (Lim, 1998). These flatworms are hermaphrodites possessing soft anatomical structures and hard sclerotized copulatory system of both males and female reproductive parts. The haptor of monogeneans consists of sclerotized armaments of one or two pairs of anchors, 14 or 16 marginal hooks and usually one or two connective bars or bars absent, which are involved in the attachment of monogeneans onto the various organs of their hosts such as gills of fishes, urinary bladder of frogs and conjunctiva of turtles and hippopotamus (du Preez & Lim, 2000; du Preez & Moeng, 2003; Lim, 1995a; Lim & du Preez, 2001). Squamous discs, clamps and suckers might also be present in the haptors of some species of monogeneans. These haptoral elements are of diagnostic importance in taxonomic description and identification of the monogenean species (Du Preez & Lim, 2000; Lim, 1995b; Lim 1996; Lim & du Preez, 2001; Lim & Gibson, 2007; Yamaguti, 1963). However, the study of these morphological structures is not easy because monogeneans are soft-bodied flatworms with fragile neodermis unable to withstand desiccation for long investigation under a microscope. The use of the digital 3D models would help taxonomist to understand the morphology as well as the functions of these morphological structures. The versatility of the digital 3D models makes them suitable and useful tools in education and research especially to elaborate on how different organs in an organism function and to elucidate the overall morphology from different viewing angles. By manipulating the digital 3D models of diagnostic features, for instance, one can ascertain whether the observed shapes and sizes are true shapes or are due to preparatory methods (Teo et al., 2010).

1.2 Research questions

The following research questions are examined in this study:

A. Is the shape deformation technique applicable in constructing 3D models of the monogenean hard parts that possess diverse form of shapes?
B. What is the feasible way to augment the existing limited training set to cater the need of the proposed machine learning algorithm used in developing an automated 3D modelling pipeline by using the 2D illustrations of monogenean as training set?
C. How to store an escalating training set for the machine learning pipeline to ease the data management work?
D. How efficient is an automated 3D modelling pipeline empowered by a machine learning algorithm in improving a deformation process of a generic 3D model into a desired 3D shape compared to the manual way of deforming a generic 3D model?

1.3 Objective of the study

The main objective of this study is to develop an automated 3D modelling pipeline that uses 2D illustration as an input to generate a variety of digital 3D models without repeating the tedious 3D modelling process from scratch. This automated 3D modelling pipeline will be driven by a machine learning algorithm that can detect the landmark points' location on an input 2D illustration and automatically align them with the vertices of a generic 3D model to enable deformation of its 3D shape to produce a target 3D model.

Specific objectives of this study are as follows:

A. To develop a digital generic 3D model for a selected monogenean hard part using a 2D illustration.
B. To develop a data augmentation technique that can generate a large amount of synthetic 2D illustration sample.
C. To customize a database to store a big dataset of 2D illustrations which will be used for machine learning.
D. To develop a machine learning model for landmark points detection assigned on a 2D illustration of a monogenean hard part
E. To evaluate the developed machine learning empowered shape deformation and 3D modelling pipeline.

1.4 Scope of the study

The development of the automated 3D modelling pipeline in this study is not aimed to construct digital 3D models for all soft anatomical and hard sclerotized structures in a monogenean. Instead, the target models of this study are limited to the haptoral anchors of monogenean from 12 selected species, viz. *Cichlidogyrus dracolemma*, *Dactylogyrus wunderi*, *Cichlidogyrus raeymaekersi, Cichlidogyrus aspiralis, Dactylogyrus primaries, Pellucidhaptor merus, Dactylogyrus falcatus, Dactylogyrus vastator, Dactylogyrus pterocleidus, Dactylogyrus falciunguis, Chauhanellus auriculatum* and *Chauhanellus caelatus*. This is because the current automated 3D modelling pipeline developed in this study is merely a prototype and its integrated machine learning model can only process the 2D illustrations of monogenean anchor to produce the corresponding digital 3D models. The development of the digital 3D models for all the anatomical structures of a monogenean is beyond the technical feasibility of this study because this will require tremendous amount of illustration training sets to train the machine learning model to enable the automated 3D modelling pipeline to produce all the required digital 3D models of anatomical structures. The current prototype version of automated 3D modelling pipeline still serves its purpose by demonstrating a working concept to produce digital 3D models for biological structures in an automated way using a machine learning algorithm. Artificial Neural Network (ANN) is chosen as the machine learning algorithm to empower the automated 3D modelling pipeline and the rationale of using ANN in this study will be presented in Chapter 3 (see Methodology).

1.5 Outline of the study

Chapter One: This chapter depicts the background and the benefits of using a digital 3D model as a research and an educational tool in biology. It is followed by the problems and limitations found in the existing digital 3D modelling methods that motivated the development of an automated 3D modelling pipeline using machine learning technique. At the end of this chapter, the research questions, objective and scope of this study are presented.

Chapter Two: This chapter presents the literature review on the monogeneans and development of deformable generic 3D models and its application in different research domains. It is followed with a discussion on the machine learning based landmark detection pipeline which includes image acquisition, landmark labelling, data storage and selection of machine learning algorithms. The relevant techniques or algorithms which were developed or used by the researchers in a landmark detection study are presented in several sub-sections respectively.

Chapter Three: This chapter presents the methodology for developing an automated 3D modelling pipeline proposed in this study and the presented details include:

A. a geometric modelling approach to develop a deformable generic 3D model of a selected monogenean anchor

B. the development of a data augmentation program to generate a big number of synthetic 2D illustrations samples for machine learning

C. the development of a no-SQL-based database to store the big dataset of 2D illustrations

D. training an Artificial Neural Network (ANN) model using the 2D illustration data retrieved from the database

E. the implementation of an ANN model to automatically detect landmark points on a 2D illustration which can direct the deformation of the generic 3D model to obtain a target 3D shape.

F. an evaluation to examine feasibility of using the shape deformation technique to create 3D models of monogenean anchor and the efficiency of the automated 3D modelling pipeline to automate a shape deformation process to derive the target 3D models

Chapter Four: This chapter presents the results obtained from the five phases of the project development which include the resulting generic 3D model of monogenean anchor, output of data preparation for machine learning, landmark point detection on the input illustrations of monogenean anchors using the machine learning model, the 3D models of monogenean anchors derived from manual deformation approach and the automated 3D modelling pipeline and lastly, the evaluation on the efficiency of the automated 3D modelling pipeline.

Chapter Five: This chapter discusses how each research questions of this study are addressed based on the research outcomes shown in Chapter 4. It is followed with the research contributions of this study. Some discussion on the constraints and limitations observed from the research outcomes are presented. This chapter ends with a conclusion on the study and some possible future development of the project.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

One limitation found in the current 3D production pipeline is the process of using existing 3D modelling tools to produce a single target 3D model is time consuming (Sandnes, 2018; Zhang et al., 2019). One will have to repeat the entire tedious modelling process for another target 3D shape from start using the 3D software. Researchers have undergone active studies on developing innovative 3D modelling approaches to circumvent this limitation and one of the alternative approaches adopted by the researchers is to derive a target digital 3D model from a pre-built digital generic 3D model by deforming its surface to form a new 3D shape (Kho & Garland, 2007; Le & Deng, 2017; Tang et al., 2014). Such approach enables direct manipulation of the 3D surface of a generic 3D model and deforms it into a variety of target shapes without repeating the tedious modelling process from start (Cheutet et al., 2005).

This chapter presents a literature review which starts with a discussion on monogenea to offer a clearer biological context shaped for this study. It is followed by the reviews on the different deformation approaches adopted by researchers to deform a digital generic 3D model to obtain a target 3D shape. Subsequently, the review on the landmark point detection studies implemented by various machine learning algorithms are discussed.

2.2 Monogeneans

Monogeneans are a group of parasitic flatworms which are mainly hosted on fishes (Lim, 1998). It possesses haptors armed with anchors, bars, marginal hooks and sometimes clamps and secretions to assist them in attaching themselves on their hosts (Figure 2.1). One main problem encountered in studying these morphological structures of the monogeneans is to visualise the specimens from dorsal, ventral, lateral as well as anterior and posterior perspectives because they are soft body and difficult to keep alive for long investigation (Teo et al., 2010). For example, it is difficult to describe and elaborate structure of a fenestrated haptor without an illustration (Lim & Gibson, 2010) and the 2D illustration alone could also hardly offer the complete visualization of how the fenestration might appear in a live worm. One viable way to resolve this visualization issue of the morphological structures in monogenean is to create digital 3D models using the 3D imaging tecniques such as confocal microscopy (Galli et al., 2006; Hodová et al., 2018) and scanning electron microscopy (SEM) (Antonelli et al., 2010; Fannes et al., 2017; Hadied et al., 2015; Poddubnaya et al., 2016). As highlighted by Galli et al. (2006), the digital 3D models of the monogenean hard parts reconstructed from the confocal microscopy can show more morphological details which are not detectable in the corresponding 2D illusrations. Nevertheless, the 3D imaging equipments are expensive and demand tedious specimen preparation (Kremer et al., 2015). Hence, an alternative 3D modelling method is proposed in this study to create digital 3D models for those morphological structures which will aid researchers in understanding the structural morphology of the monogenea. The following section will present the literature review on the alternative 3D modelling methods by focusing on shape deformation tecniques.

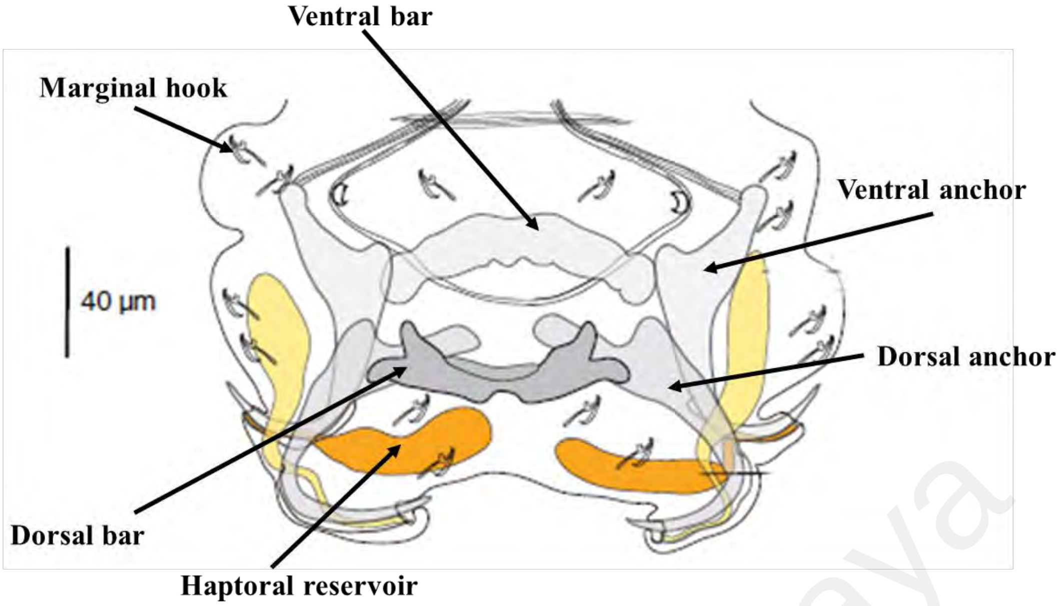


Figure 2.1: Haptor of *Sundatrema langkawiense* (a monogenean). Adapted from Lim & Gibson (2009).

2.3 3D modelling based on the deformation of a generic 3D model

The deformation of a digital generic 3D model approach is well accepted in the human face modelling because of the need for an effective way to model various human facial expression (Ansari et al., 2003). The core idea behind this modelling approach was to deform a pre-built 3D head model mesh to match the facial expression given by either one source image (Hähnel et al., 2006; Moeini et al., 2013; Zhang et al., 2015) or multiple source images captured from the front view and the side views of a target human face using a camera (Lee et al., 2001). The deformation of the generic 3D face was achieved by aligning the selected vertices of the generic face to match with their corresponding landmark points which were manually assigned on a 2D source image (Huang et al., 2016) or were estimated through a probabilistic distribution (Wang & Lai, 2008). This deformation of a generic 3D model approach shows the capability to produce a variety of vivid facial expression.

The application of similar 3D modelling approach was also found in some 3D modelling works for other human parts such as a hand (Rhee et al., 2006), foot (Zhao et al., 2015) and tooth (Zheng et al., 2011). However, the deformation of the generic models of those human parts need to deal with a higher variation of shapes with many degrees of freedom such as the movement of a hand and a foot (Rhee et al., 2006; Zhao et al., 2015) or an occlusal surface found in a teeth (Zheng et al., 2010). Instead of matching the vertices of the generic 3D model to the corresponding landmark points on the 2D image, Rhee et al. (2006) resorted to extract the curve segments from a generic 3D hand and align them with the contour lines of a hand in a 2D source image to deform the generic model into a target shape. In the teeth modelling work presented by Zheng et al. (2011), a Snake model algorithm adapted from two previous studies (Kass et al., 1988; Williams & Shah, 1992) was used to automatically identify the salient landmark points on a scanned 2D source image of a teeth surface and the vertices of a digital generic 3D teeth model were aligned with these landmark points to deform the generic shape into a target model through an auto-mapping algorithm based on a radial basis function. As for human foot modelling, Zhao et al. (2015) presented a very different deformation approach by directly matching the selected vertices on a generic 3D model of the human foot to their corresponding vertices on another 3D skeleton foot model taken from an existing 3D database in OpenSim, a 3D musculoskeletal modelling system developed by Delp et al. (2007). The generic 3D foot model was deformed by moving the vertices on the 3D skeleton foot and the degree of the deformation was controlled by OpenSim. OpenSim was able to calculate the precise degree of deformation of a generic 3D human foot model to ensure the shape deformation was following a physical constraint imposed by the muscle system inside a human foot. This produces a highly accurate 3D model of the foot for a scientific study rather than just an approximated or arbitrary 3D shape.

The deformation of a digital generic 3D model approach is also not only limited to modelling the human parts but also the entire human body (Starck & Hilton, 2008) and 3D animation characters (Baran & Popović, 2007; Blanco & Oliveira, 2008; Min et al., 2009). One typical difference observed from these studies is that the deformation of a generic 3D model of human or animation character body relied on a skeletal rigging system embedded within the generic model such as the one shown in Figure 2.2. This skeletal rigging system functioned as a handler to deform the shape of the digital generic 3D model by moving the control points that compose the rigging system.

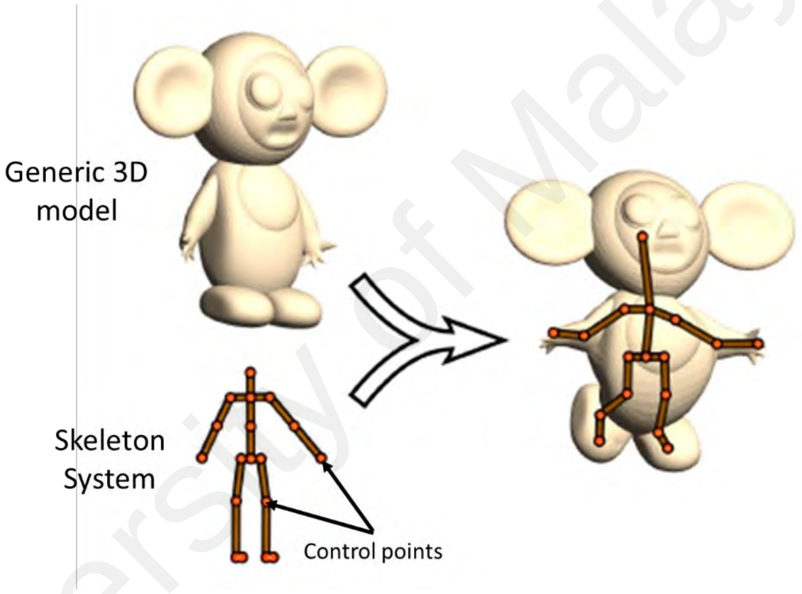


Figure 2.2: A digital generic 3D model of a cartoon character embedded with a skeletal rigging system inside it. Adapted from Baran & Popović (2007).

Blanco and Oliveira (2008) showed an intuitive way to create a skeletal rigging system by allowing a user to merely sketch selected target parts of the digital generic models and each of the sketched line was hereafter converted to a 3D parametric curve that could be used as a handle to deform the generic model (Figure 2.3). The skeletal rigging system can offer a smooth deformation of a digital generic character model but with some physical constraints that could prevent distortion of a 3D shape.

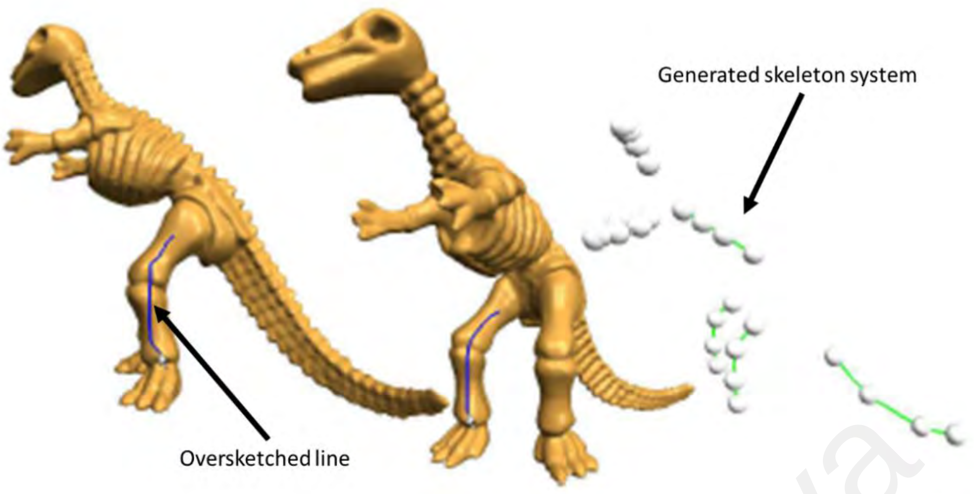


Figure 2.3: An intuitive way to over sketch a line onto a target part of the digital generic model to generate a skeleton system which functioned as a deformation handler. Adapted from Blanco & Oliveira (2008).

On the other hand, there was a relative paucity of records reported on using the deformation of a generic 3D model approach to construct a 3D model for biological specimens. One example is the work presented by Murakawa et al. (2006) that developed 3D models for insects. It is worth noting that Murakawa et al. (2006) demonstrate an interactive way to enable a user to draw a stylus lasso on a selected area of a 2D source image which would then be converted to a sequence of landmark points (termed as sketch points) on the boundary of the 2D shape. The control vertices of a digital generic 3D model of insect were then aligned with those landmark points and deform the generic shape into a target 3D model.

In summary, many modelling works as aforementioned involved an alignment between the vertices of a generic 3D model with their corresponding landmark points assigned on one or more 2D source images. These works presented an important idea that the deformation of a generic model is dependant on tracking the landmark points assigned on the 2D source image to direct the deformation of the 3D surface into a desired shape.

Those landmark points were either manually assigned or estimated using some complex mathematical or computer vision algorithms such as the Snake algorithm which might not be a trivial task to a researcher without computer graphic skills to apply or reproduce it to develop a 3D modelling pipeline for another target group of objects such as monogenean hard parts. With the growth of computing power and storage in recent years, the landmark detection on a 2D image input could be automated by a machine learning technique (Ranjan et al., 2018). The section below will provide further discussion on the recent development of landmark points detection on a 2D image implemented by machine learning algorithms.

2.4 Landmark points detection using machine learning algorithms

Landmark point detection has been studied extensively by researchers for tracking human facial feature points which were applied to human face recognition (Chen et al., 2017; Ranjan et al., 2017; Zhang et al., 2016), facial expression analysis (Gondhi et al., 2017), facial landmark localization (Deng et al., 2018; Fan & Zhou, 2016), age estimation (Zaghbani et al., 2018), gender classification (Jia & Cristianini, 2015) and 3D face modelling (Ferrari et al., 2017). These studies share a common idea that the facial feature points located at the eye, nose, mouth and chin of a 2D facial image carry semantic meaning and these points could automatically be detected via a supervised or unsupervised machine learning process using annotated 2D sample images as a training dataset. Besides, such similar idea of the landmark detection was also applied to several other fields of study such as geometric morphometric for bioimages (Bromiley et al., 2014; Vandaele et al., 2018), anatomical features detection for medical diagnosis (Prentašić & Lončarić, 2016; Zhang et al., 2016), and human body pose detection (Lovato et al., 2014). In general, the entire procedure of landmark point detection based on a machine learning algorithm includes 2D sample image acquisition, landmark annotation, data storage and selection of a machine learning algorithm to train a machine learning model (Figure 2.4). At the end, the machine learning model is used to automate the landmark point detection by fetching an image input to the model.

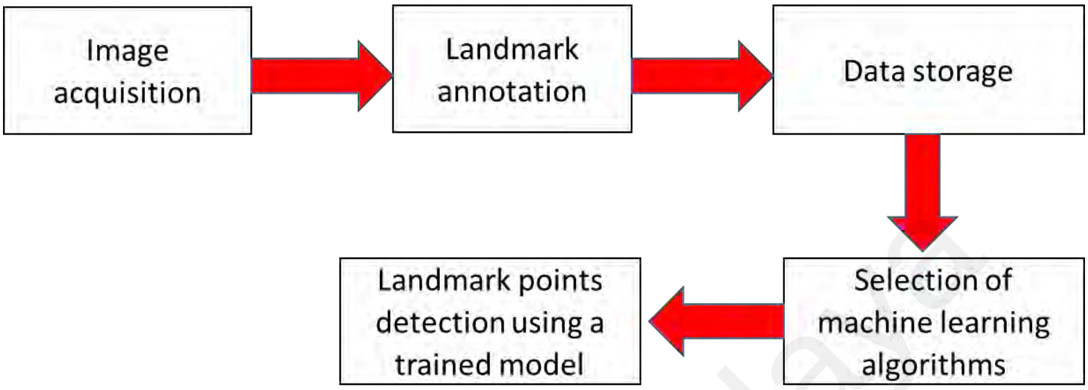


Figure 2.4: General pipeline of a landmark point detection based on a machine learning algorithm.

The following sub-sections will present in more details each step in a landmark point detection pipeline reported in different studies.

2.4.1 Image acquisition

The aim of image acquisition is to obtain the source images as a training set for machine learning. Generally, the methods of image acquisition found in the reviewed literature could be differentiated based on the image sources such as the ones taken from a camera (Baltrušaitis et al., 2016; Jeong et al., 2017), a computed tomography scanner (Le-Tien & Pham-Chi, 2014; Wang et al., 2015; Zhang et al., 2016), a video sequence (Chen et al., 2017) or an existing image database (Fan & Zhou, 2016; Ranjan et al., 2016; Vandaele et al., 2018). In some studies, preprocessing of these source images was done to resize them in a uniform aspect ratio (Gondhi et al., 2017; Martinez & Pantic, 2015) as many machine learning algorithms assume a square shape input image for an optimum performance. In some other studies, the source images were also turned into the grayscale images (Jia & Christianini, 2015) or binary images (Mondéjar-Guerra, 2018) to reduce the computational cost.

It is also worth noting that a common problem encountered by some researchers during the image acquisition stage is the absence of a sufficient dataset for training an optimum machine learning model and this usually leads to an overfitting issue. As stated by Chollet (2017), a model shall be trained on more data to obtain a better generalized machine learning model with minimized overfitting effect. In response to this problem, some researchers included data augmentation in their studies to increase the existing image dataset by flipping, translating and cropping the images (Farfade et al., 2015; Ren et al., 2016; Xu & Kakadiaris, 2017). Krizhevsky et al. (2012) also reported another form of data augmentation by altering the intensities of the RGB channels in the training images. These data augmentation approaches are collectively termed as oversampling (Wu et al., 2017) which could greatly increase the training and testing dataset for a machine learning study.

2.4.2 Landmark annotation

Landmark annotation is a procedure where a set of markers are assigned to some selected point of interest (POI) on a training dataset of 2D images. This set of markers are known as landmark points/feature points and they carry semantic meaning which enables them to be an important feature for training a machine learning model in later stage (Wang et al., 2018). One simple and direct landmark annotation method is to select some POIs on the image samples and manually label those POIs with some markers. This manual annotation approach is found in the landmark detection study for geometric morphometric (Bromiley et al., 2014; Vandaele et al., 2018) and craniofacial landmark localization (Shahidi et al., 2014). The selection of those POI was verified by some domain experts. Figure 2.5 shows an example of manual landmark annotation demostrated by Vandaele et al. (2018).

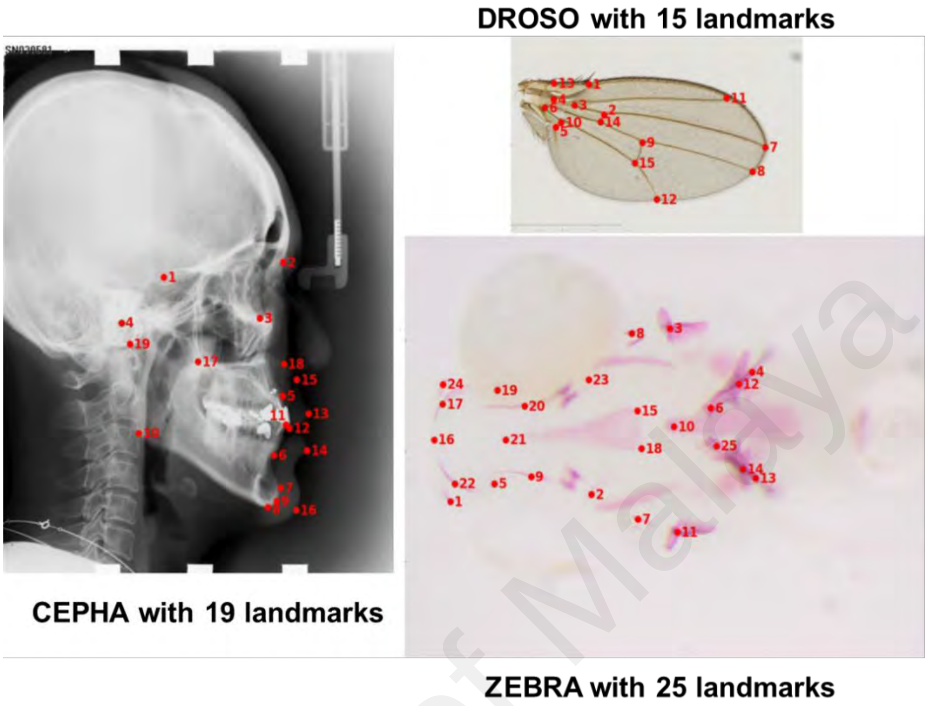


Figure 2.5: Sample image and corresponding landmarks for each dataset.

On the other hand, Sagonas et al. (2013) argued that it is not a trivial task to manually annotate landmarks for a big dataset of images as it requires trained human for this job and it is also a laborious and error-prone process. In an earlier study, Tong et al. (2012) had developed a semi-supervised approach by using only 15 manually annotated human face images as the input to automatically estimate the landmark point locations on a complete set of 1176 human face images through the Semi-supervised Least-Squares Congealing (SLSC) algorithm. Such semi-automated approach greatly improves the efficiency of annotating the dataset to train a machine learning model. While the semi-supervised method could significantly expedite a landmark annotation process, many researchers tend to adopt another alternative appproach by using the pre-annotated image samples taken from the existing open data as their training set and this practice is especially common in many human facial landmark detection studies (Cevikalp et al., 2013; Deng et al., 2018; Fan & Zhou, 2016; Ferrari et al., 2017; Gondhi et a., 2017; Köstinger et al., 2011; Milborrow et al., 2010; Ranjan et al., 2017; Sagonas et al., 2013; Sánchez-Lozano et al., 2018; Wu et al., 2017). Open data is dataset which is freely available to any party such as scientists seeking to exploit it and add value to it (Sedkaoui & Monino, 2016).

Although the usage of the pre-annotated images obtained from the open data could save the researchers from the labourous task of annotating every individual image samples for a landmark detection study, this approach is only applicable to a specific researh domain where the open data is relevant with the study. This highlights the importance of developing a storage system which is dedicated to a specific research domain for both current and future studies which can stimulate data reusability and prevent reinventing the wheel. The following section presents further discussion on the development of data storage system in the landmark detection studies.

2.4.3 Data storage

In landmark detection study, many researchers are still using a conventional approach of data storage in file-based system. In many cases, the image samples were kept in one or more folders whereas the annotated landmark data and some other associated parameters were kept in other separate text files or spreadsheets (Aifanti et al., 2010; Belhumeur et al., 2013; Gross et al., 2010; Le et al., 2012; Martínez & Benavente, 1998; Messer et al., 1999; Milborrow et al., 2010; Nordstrøm et al., 2004; Shahidi et al., 2014; Tonsen et al., 2016; Vandaele et al., 2018; Zhu & Ramanan, 2012). These file systems store the image samples which were taken either from a real world environment or a controlled environment and their image data varied from one to another based on the image resolution, image type and number of landmarks per sample image. Some file systems were given a specific abbreviated name by their authors and the annotated landmarks on each image sample were represented as the 2D Cartesian coordinates.

Although file system is a simple and portable storage medium, it is less structured and the data management becomes very complicated when the size of data escalates over time. A better approach would be to develop a structured database. A database is an organised and formal collection of information stored in a computer readable format and this enables the dataset to be structured in a more formalised manner while the relationships between the data items remain meaningful and consistent (Dhillon S.K, 2018). The development of a database for a landmark study is relatively scarce and one relevant work is featured by Köstinger et al. (2011) where they developed a relational database using a database management system (DBMS), SQLite, to keep their annotated landmark data and the directory path where the images samples were kept. Their SQLite database was able to facilitate the integration of image samples taken from different collection sources and also offered a convenient way to perform a data query by using few simple SQL commands.

Nevertheless, a relational database like SQLite is only capable of handling medium-size datasets (Varian, 2014) and it will become cumbersome to manage a dataset with several gigabytes of size such as the MultiPie dataset (Gross et al., 2010). An alternative database model which is capable of scaling a large dataset is the NoSQL database (Ercan, 2014). The architecture of a NoSQL database is designed to overcome the limited scalability, flexibility, performance, availability and infrustructure cost issues associated with the relational database (Stonebraker, 2010). Recently, NoSQL database started to receive much attention as the demand increases for high speed data access over large volume of data without much effort in scaling and tuning (Dhillon, 2018). In this study, a NoSQL database was developed to store the images and their associated landmark data. The details of the NoSQL database development will be presented in Chapter 3.

2.4.4 Common machine learning algorithms for landmarks detection

Machine learning, a term coined by Samuel (1959) who defined it as a field of study that gives computer the ability to learn without being explicitly programmed. The core idea behind this machine learning is to fetch data into a computer to enable it to learn from data (Géron, 2017). Over the years, researchers have developed various machine learning algorithms such as Artificial Neural Network (McCulloch & Pitts, 1943) Support Vector Machine (Cortes & Vapnik, 1995), Random Forest (Ho, 1995), Decision Trees (Quinlan, 1999) and etc. These machine learning algorithms were applied in many different domain of studies and one of them was landmark detection study (Khalid et al., 2014; Ranjan et al., 2016; Wang et al., 2018). The selection of a machine learning algorithm is very much dependant on the research question addressed by the researchers which is either a classification or regression problem. Another two factors that affect the selection of a machine learning algorithm are the volume of available dataset and also the availability of the computing power. In the following sections, some common machine learning algorithms which were reported in the reviewed literatures are discussed further.

2.4.4.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a numerical classifier that attempts to draw a straight decision boundary line between the classes of data points, creating regions where all points inside that region will be considered as a member of that class (Cortes & Vapnik, 1995). The boundary lines not only separates the two classes but also stays as far away from the closest training instances as possible which are located on the margin lines on both sides (Figure 2.6).

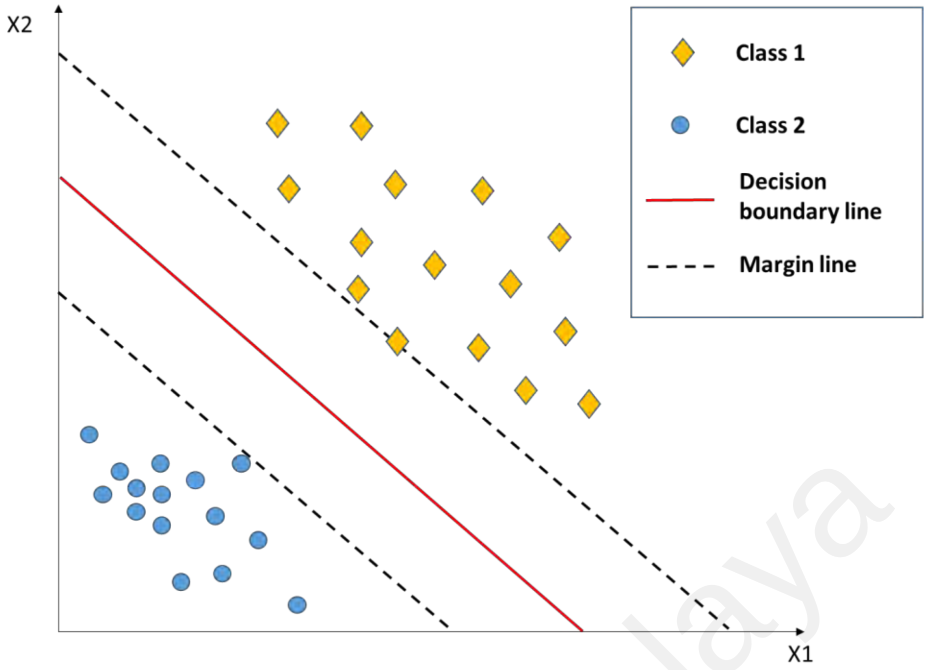
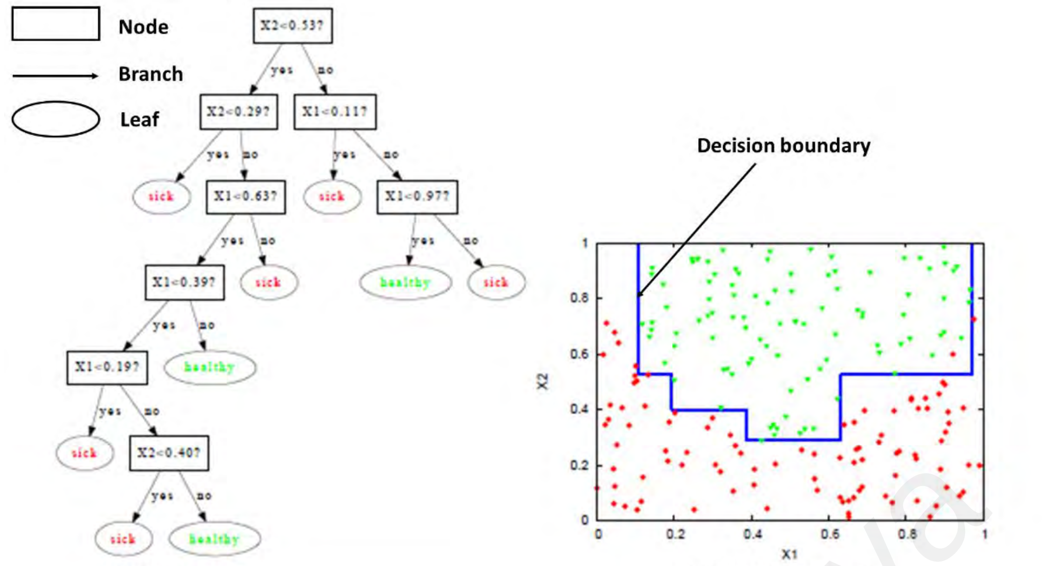


Figure 2.6: A linear SVM classification.

SVM was adopted by Samara et al. (2017) to train a number of binary classifiers with each of them corresponded to a human facial emotional state by using coordinates of the facial landmarks as their training set. The resulting binary classifiers were used to compose a Hierarchical Parallelised Binary Support Vector Machines (HPBSVM) model which was then used to predict a human facial expression based on the combination of the decisions given from the individual binary SVM classifier component. Cevikalp et al. (2013) had also used a cascade of SVM binary classifiers to detect the landmark points position on a human face. In their work, two types of detectors were trained which were roots and parts detectors where the roots detector returns the candidate image regions that cover the entire face, and the parts detector searches for landmark locations within the candidate region. However, the SVM classifiers developed by Cevekalp et al. (2013) could only perform well if the clarity of the given face images are good and the image size is big. Such limitation of SVM classifiers under the challenging image conditions was addressed by Mondéjar-Guerra et al. (2018) who produced an exclusively synthetic image dataset to train a learning model. The synthetic images were generated by applying some random transformations such as motion blur, dynamic range compression, affine transformation, non-uniform lighting, and dilation onto every training image dataset. The SVM model trained from those synthetic images was able to detect landmark locations under a poor image condition. The application of SVM is also not limited to facial landmark detection but is also found in a cephalometric landmark detection study (Le-Tien & Pham-Chi, 2014). The cephalometric landmarks were detected from a region of a cephalogram input using the combination of the Histograms of Oriented Gradients (HOG) descriptor with the SVM classifier.

2.4.4.2 Decision Tree and Random Forest

Decision Tree and Random Forest are two machine learning algorithms which are closely related to each other as the Decision Trees are the fundamental components of a Random Forest (Ho, 1995; Quinian, 1999). Prior to introducing Random Forest and its application in some landmark detection studies, it is necessary here to briefly describe the Decision Tree. A Decision Tree is a tree-like decision model where each node represents a feature (attribute), each branch represent a decision or rule and each leaf represent an outcome which is either a categorical or a continous value. A Decision Tree divides the feature space into a axis-parallel rectangle which is known as a decision boundary (Figure 2.7).



(A) Decision tree

(B) Decision boundary

Figure 2.7: A decision tree and its corresponding decision boundary. Adapted from Geurts et al. (2009).

Although Decision Tree is famous for its simplicity in constructing, it still demands a user's expertise to interpret and validate its prediction outcome (Zorman et al., 1997) and its accuracy level of prediction is not as competitive as with other machine learning algorithms (Malekipirbazari & Aksakalli, 2015). On the other hand, it is possible to train a group of decision tree classifiers using different subset of the traning data. The resulting ensemble of Decision Trees is known as a Random Forest (Figure 2.8). The final predicted outcome is aggregated from the predictions given by each individual Decision Tree in the Random Forest via a majority voting mechanism (Ho, 1995). The Random Forest can give a prediction which has a much higher accuracy compared with a single Decision Tree.

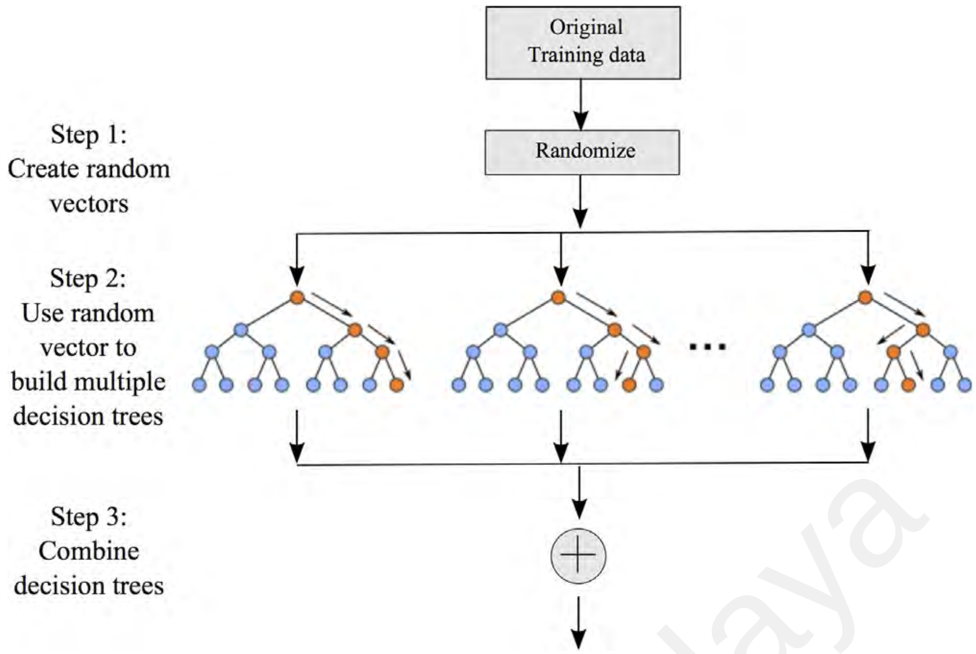


Figure 2.8: Random forest methodology. Adapted from Malekipirbazari & Aksakalli (2015).

The application of Random Forest algorithm was reported in the landmark detection for medical analysis (Criminisi et al., 2011; Wang et al., 2015), human facial analyis (Dantone et al., 2012; Jia et al., 2014) and geometric morphometrics (Bromiley et al., 2014; Vandaele et al., 2018). The training data used to build the Random Forest models in those studies was either the coordinates of the anatomical landmarks or the facial landmarks which were manually annotated on the training images by the researchers. The resulting Random Forest models were used to estimate the probability or the coordinate location where a target landmark was detected on an input image. In a more advanced setting presented by Jia et al. (2014), the Random Forest models were trained for both regression and classification tasks. Their Random Forest models were able to classify a face or non-face region on an input image (a classification task) and at the same time to locate the coordinates where the facial landmarks were detected on the input image (a regression task). Dantone et al. (2012) also proposed a conditional Random Forest framework for their human facial lanmark detection study in which one Random Forest which consists of multiple Random Forests were trained using the subset of their training data illustrated by the head poses. When testing on a facial image, the head pose was first detected and one corresponding Random Forest member was then selected to estimate the coordinates of the facial landmark points. The usage of this conditional Random Forests could greatly enchance the accuracy of a landmark detection and at the same time still maintain a low computational cost (Sun et al., 2012).

2.4.4.3 Artificial Neural Network

Aritificial Neural Network (ANN) is a simplified computational model inspired by the biological neuron network which is able to perform complex computations using propositional logic (McCulloch & Pitts, 1943). A simple and well-known ANN architecture known as the Perceptron was proposed by Rosenblatt (1958). A Perceptron is composed of a single layer of linear threshold unit (LTU) which includes a set of input units known as neurons, an activation function and an output unit (Figure 2.9).

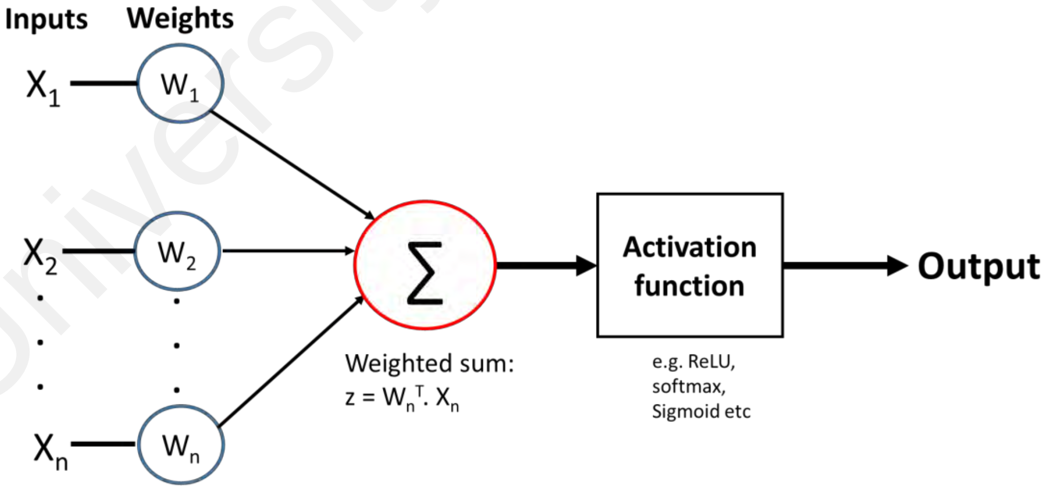


Figure 2.9: Architecture of a Perceptron.

A LTU computes a linear combination of the inputs by multiplying each input by a weight and sums the weighted input's values. The summed value is fed to an activation function and if the resulting value exceeds a threshold, it returns an output value. Over the years, researchers have developed different activation functions for ANN studies and many of them are derived from the mathematical studies. The output value is dependant on the activation function adopted in a particular ANN system which can vary between 0 and 1, -1 and 1 or 0 and infinity (Table 2.1). The selection of the activation funtion in an ANN system depends on the nature of a particular study which can be a binary class classification, a multi-class classification or a regression problem.

Table 2.1: Some popular activation functions used in ANN.

Activation function	**Range of output values**	**Reference**
Softmax	0 to 1	Sutton & Barto, 1998
Sigmoid	0 to 1	Han & Moraga, 1995
ReLU	0 to infinity	Nair & Hinton, 2010
Leaky ReLU	0 to infinity	Maas et al., 2013
Tanh	-1 to 1	Spanier & Oldham, 1987

Practically, the training of an ANN always involves a Multi-Layer Perceptron (MLP) instead of a single Perceptron. An MLP consists of one input layer of neurons, one or more layers of LTUs which are known as hidden layers and one output layer (Figure 2.10).

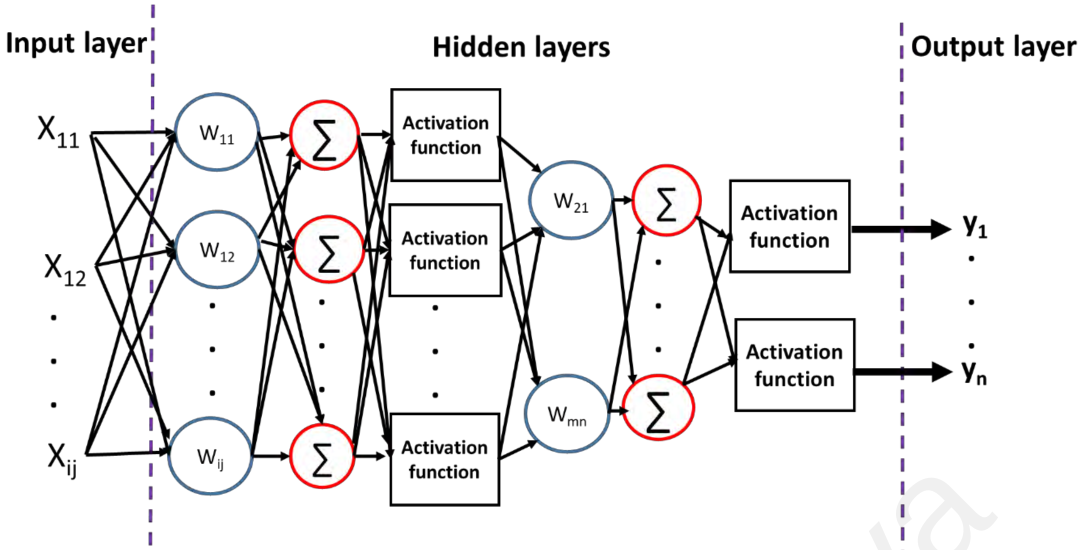


Figure 2.10: An architecture of Multi-Layer Peceptron.

The training of an ANN involves two processes which are forward propagation and backpropagation. In the forward propagation, the feature values from the input layer propagate through the hidden layers where the feature values will be processed successively by the LTUs with the initialized weight parameters which are usually a random set of small values from a Gaussian or normal uniform. The predicted values from the last hidden layer will be sent to the output layer and will be compared with a corresponding set of ground truth values using a loss function. The difference between the predicted and ground truth values is termed as loss. In the backpropagation process, the ANN propagates backwards from output layer to the hidden layers to identify how much each weight parameter contributed to the loss. At each weight parameter, an optimizer such as Stochastic Gradient Descent algorithm or Adam algorithm is used to update the parameter value to reduce the loss. When training ANN, the process of forward propagation and backpropagation are repeated to keep tuning the weight parameters in the ANN until an optimum prediction is obtained. This entire training process that involves repeated mapping of input feature values to the ground truth values as described above is also known as supervised learning (Shukla, 2018). There is another type of training method which enables an ANN to automatically learn the inherent structure of the input feature values without using the explicitly-provided ground truth values. Such training method is known as unsupervised learning (Shukla, 2018).

In landmark detection study, the most common form of ANN used by the researchers was Deep Neural Network (DNN). A DNN consists of two or more hidden layers in its network architecture and it is capable of capturing small details of the input features and generalize them exceed any expectations (Bonaccorso, 2017). Apart from the standard MLP model, there are also several variants of DNN architectures reported in the literatures such as the Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Autoencoders. Each of the DNN was trained using a supervised, unsupervised or combination of both learning methods.

Generally, the application of DNN such as CNN, RNN and autoencoder were generally reported in the more recent studies and the most common one was CNN. The popularity of CNN in recent years is due to its great effectiveness for image analysis tasks (Krizhevsky et al., 2012). In some studies, CNN was used along with a clustering algorithm to partition the training sets into several subsets which share the common properties (Wu et al., 2017) or with autoencoder to rectify the outlier points (Deng et al., 2018) and such coupling approach was able to enhance the prediction accuracy in the CNN. Autoencoder was also used on its own to automatically identify human facial expressions using some geometrical features and the features data was trained with an unsupervised method. On the other hand, Chen et al. (2017) used RNN in their landmark detection study to deal with the non-linear deformations in the human facial shape.

CHAPTER 3: METHODOLOGY

3.1 Introduction

As stated in Chapter 1, the aim of this study is to develop an automated 3D modelling pipeline empowered by a machine learning algorithm to construct 3D models of monogenean anchor in an automated way. The monogenean anchors were selected as the target models because they are important attachment appendages for monogeneans. The shape and size of the anchors are taxonomical informative (Khang et al., 2016) and are useful for systematic and geometric morphometric study (Lim, 1986; Lyakh et al., 2017; Rodríguez-González et al., 2017; Tan & Lim, 2009). There were a total of 12 monogenean anchors selected as the target models and all of them were constructed by deforming a digital generic 3D model of anchor using the automated 3D modelling pipeline empowered by a machine learning algorithm. Prior to presenting the development of the entire automated 3D modelling pipeline empowered by the machine learning algorithm, this chapter will first present the development of the digital generic 3D model of anchor. This digital generic 3D model will be used as a 3D template to derive another target 3D shapes in later stage. The construction of digital generic 3D model was followed by data preparation to train a machine learning model which will be used for landmark detection. The machine learning model was trained with 2D illustrations of the monogenean anchor through a supervised learning process. Due to the scarcity of available 2D illustrations of monogenean anchors which could be taken from publications to cater the need of a machine learning model, a 2D data augmentation program was developed to synthesize the illustration training sets. Subsequently, a NoSQL database was developed and was used as a data management system to store all the augmented training sets. These training sets were then used to train a machine learning model which can detect landmark points on an input 2D illustration. At the end, a 3D modelling system was developed and was integrated with the machine learning model to implement the automated 3D modelling pipeline proposed in this study. The digital generic 3D model developed in the previous step was loaded into 3D modelling system and this modelling system was then used to automatically deform the generic model into the target 3D shapes by fetching the input illustrations of target anchor to the system. The automated deformation was achieved by using the landmark points predicted by the machine learning model to direct the deformation of the digital generic 3D model into diverse forms of the desired 3D shapes. The resulting 3D models produced from the automated 3D modelling pipeline were compared with the corresponding 3D models produced using a manual deformation approach. At the end, the efficiency of the proposed automated 3D modelling pipeline was examined by creating the boxplots and by calculating the Euclidean distance between the landmark points predicted by the machine learning model and their corresponding control points which were annotated on every single input 2D illustration. The entire project flow is illustrated in a diagram below (Figure 3.1). The project flow was divided into five phases and each phase is also tied with one project outcome based on the specific objectives defined in Section 1.3.

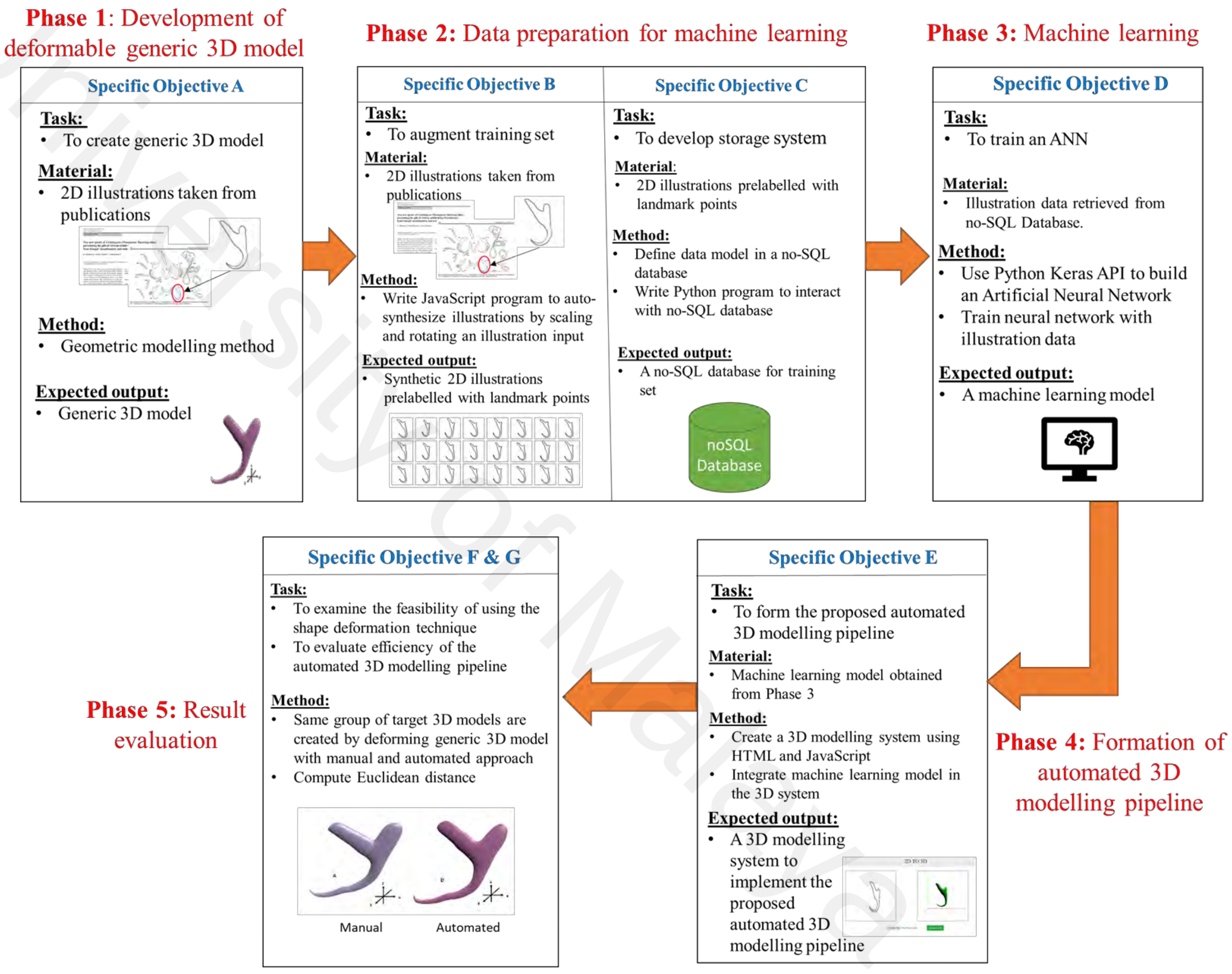


Figure 3.1: Project flow.

3.2 Phase 1 – Development of a digital generic 3D model

In this section, the procedure of building the digital generic 3D model of monogenean anchor is presented in detail. The development of the digital generic 3D model was based on a geometric modelling approach which used a stack of rectangular building blocks to form the 3D shape of the generic 3D model. The procedure can be broken down into a sequence of steps as follows:

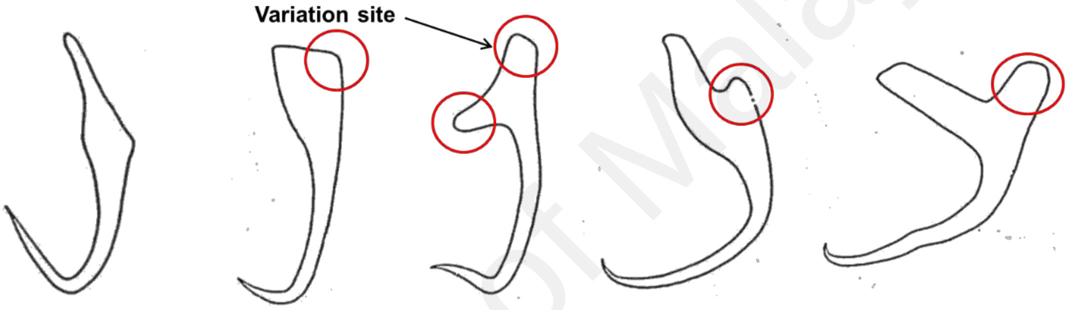
A. Acquisition of 2D illustration of monogenean anchor as 2D template
B. Assigning point primitives on the 2D template
C. Constructing 3D wireframes for preliminary generic 3D model using Mathematica and Blender
D. Constructing 3D rectangular building blocks and developing preliminary generic 3D model in Blender
E. Optimization of number of point primitives to cater for deformation
F. Formation of final generic 3D model

The following sub-sections will depict each of the steps to construct a digital generic 3D model by using geometric modelling approach.

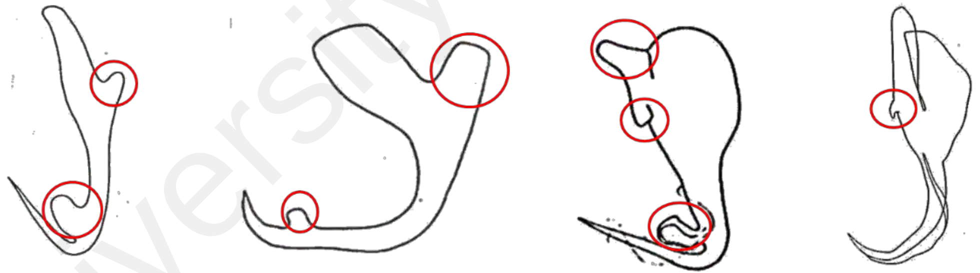
3.2.1 Acquisition of 2D illustration of monogenean anchor

Prior to starting the process of developing a digital generic 3D model, a 2D illustration of an anchor of a simple dactylogyridean monogenean was selected to be used as a 2D template. This sample was selected by reviewing the different publications on monogeneans species belonging to the dactylogyrideans to extract all possible monogenean anchors types (Figure 3.2) (see Gusev, 1976; Lim, 1994). After analyzing the different anchors, the morphologically simple anchor of *Bivaginogyrus obscurus* Gusev, 1955 was selected as the 2D template (Figure 3.2A).

It is also necessary that the digital generic 3D model be deformed into all possible morphological diversities. Therefore, these anchors were also examined to determine the locations or sites of high morphological variations such as “bulges”, ornamentations and “extrusion” (Figure 3.2B-3.2I). Seven sites were noted on the anchors where most morphological variations are likely to occur. This information will be used in optimization process (see Section below).



(A) ***Bivaginogyrus obscurus* Gusev, 1955**
(B) ***Dactylogyrus primarius* Gusev, 1955**
(C) ***Pellucidhaptor merus* Zaika, 1961**
(D) ***Dactylogyrus Falcatus* Wedl, 1857**
(E) ***Dactylogyrus* Nybelin, 1924**



(F) ***Dactylogyrus pterocleidus* Gusev, 1955**
(G) ***Dactylogyrus falciunguis* Achmerow, 1952**
(H) ***Chauhanellus auriculatum* Lim, 1994**
(I) ***Chauhanellus caelatus* Lim, 1994**

Figure 3.2: 2D illustrations of selected monogenean anchors from Gusev (1973) and Lim (1994).

3.2.2 Point primitive assignment on the 2D template

The step of assigning point primitives on the 2D template was attempted using two different approaches. The first approach relies on a conventional way to manually assign a set of point primitives along the selected 2D anchor outline which was printed on a Cartesian graph paper. (Figure 3.3). More point primitives (instead of 4 point primitives at normal sites) were added and stacked at the 7 sites noted to have high morphological variations in the anchor to allow for more deformation of the final generic 3D model. The Cartesian graph paper allows each of the primitive point to be manually defined by a set of Cartesian coordinates which are important parameters to construct rectangular building block for 3D model.

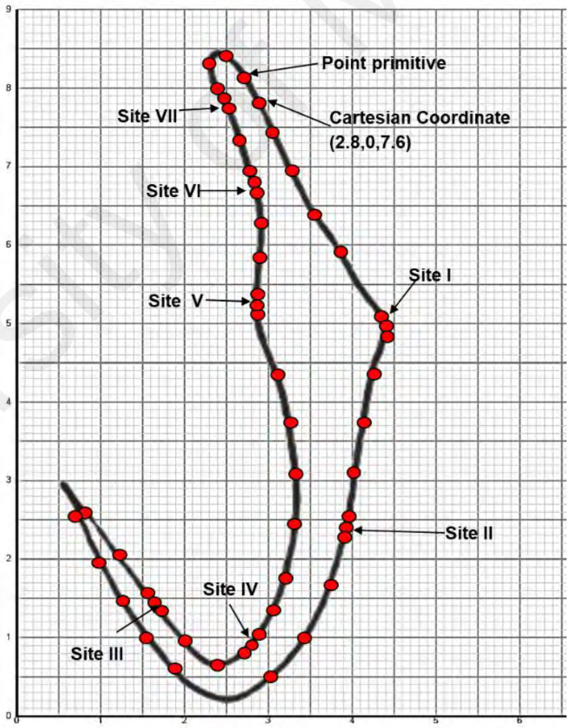
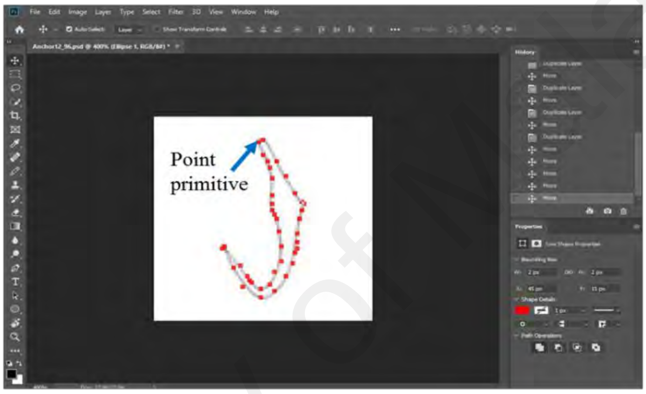
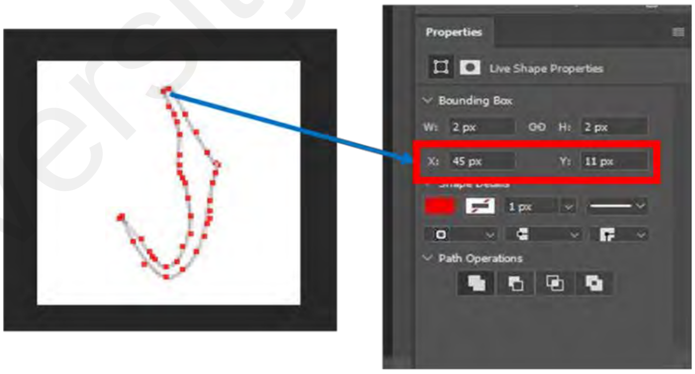


Figure 3.3: The first approach of point primitive assignment on 2D template printed on a Cartesian graph paper.

As presented above, the first approach involved a process of extracting coordinates for each assigned point primitives from the graph paper. This extraction process is tedious and therefore a second approach that can achieve the same aim was attempted. The second approach was done by uploading the same 2D illustration of anchor onto Adobe Photoshop and then the similar process of assigning point primitives along the 2D outline was repeated using the Eclipse tool in the Photoshop (Figure 3.4A). The Adobe Photoshop enabled an auto-extraction of coordinates for each of the point primitives assigned onto the uploaded 2D anchor (Figure 3.4B). The coordinates of these point primitives were collected from the Properties window in the Adobe Photoshop and they were used as the raw data to build a 3D wireframe of the generic 3D model as depicted in following section.



(A) Upload 2D template onto Adobe Photoshop



W: 2 px H: 2 px

X: 45 px Y: 11 px

1 px

(B) Extract the coordinates from Properties Window in Adobe Photoshop

Figure 3.4: The second approach to assign point primitives on the 2D template.

3.2.3 Construction of 3D wireframe for preliminary generic 3D model

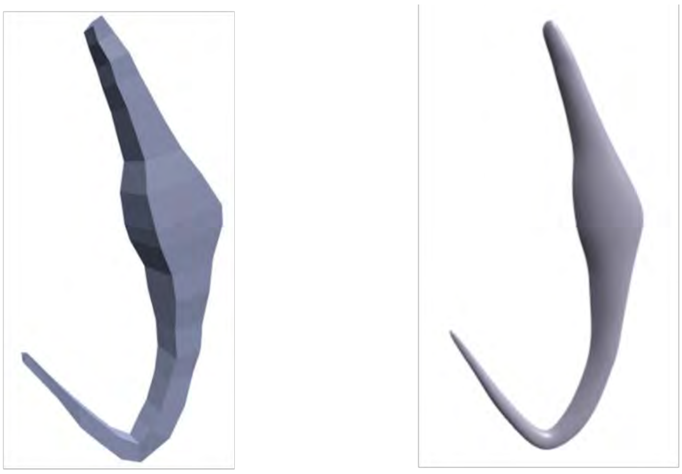
The construction of the wireframe is to visualize the initial underlying structure of the preliminary generic 3D model by displaying its rectangular building blocks (skeleton). The wireframe of the preliminary generic 3D model were constructed using Blender

(Figure 3.5). The Cartesian coordinates of the point primitives obtained from the Adobe Photoshop were used as parameters to develop the wireframe for the preliminary generic 3D model.

Figure 3.5: 3D wireframe of preliminary generic 3D anchor constructed in Blender.

3.2.4 Construction of 3D rectangular building blocks and preliminary generic 3D model development

A Python script was written using the Python Application Programming Interface (API) in Blender to construct 3D rectangle as the building blocks for the preliminary generic 3D model. The Cartesian coordinates for each rectangular block were used as parameters in the Python scripts to form the 3D geometry of the anchors (Figure 3.6A). The Catmull-Clark subdivision surface modifier, a smoothening tool in Blender, was used to smoothen the angular 3D geometry of the resulting 3D model. Jagged edges along the newly developed 3D model were reduced by rendering it using anti-aliasing option in Blender to obtain a smoother preliminary generic 3D model of anchor with less jagged edge (Figure 3.6B). However, these jagged edges could not be completely eliminated.



(A) Before smoothening

(B) After smoothening using Catmull-Clark subdivision surface modifier and anti-aliasing rendering option

Figure 3.6: Preliminary generic 3D model of anchor.

3.2.5 Point primitive for 3D deformation

The basic number of point primitives for a rectangular surface is four. However, in order to ensure that proper deformation of the generic 3D model to different diverse shape there is a need to increase the number of point primitives at certain sites on the model especially at sites of high morphological variations (Figure 3.3). Nevertheless, it is important to balance the number of point primitives because too many point primitives will make deformation tedious and complicated as these points are then required for the deformation process. Hence, before producing the final generic 3D model, an optimal number of point primitives were required at sites of high morphological variations.

In order to optimize the number of point primitives, all the known diverse anchors were first examined (Figure 3.2) to determine the degree of variation a particular site is likely to have (see section 3.2.1). The preliminary generic 3D model (Figure 3.6B) developed in Blender without any additional point primitives was used in the optimization of point primitives. In this optimization process the target anchor used was *Dactylogyrus pterocleidus* Gusev, 1955. The target anchor was imported into Blender (Figure 3.7A) and the preliminary generic 3D model was aligned onto it for deformation by manual manipulation to derive the 3D anchor (Figure 3.7B). The number of point primitives was increased two-fold at Site IV till it is able to be deformed into outer root (Site IV) to obtain the desired target shape and size (Figure 3.7D & 3.7F). The minimum numbers of point primitives necessary to deform each of the sites to attain the most diverse shape is given in Table 3.1.

The preliminary 3D model with 4 point primitives at site IV could not be deformed to obtain the desired shape of the outer root of the 3D anchor model for *Dactylogyrus pterocleidus* (Figure 3.7C). There were insufficient point primitives to deform to fit the target shape (Figure 3.7C). In developing the outer root, 8 point primitives were needed at Site IV to obtain the outer root although the outer root is morphologically a simple structure (Figure 3.7G). The same process was done for all the other 6 sites. As expected, the sites having higher morphological variations have higher number of point primitives (Table 3.1). For example, at Site VI, 16 point primitives were needed to deform and create the ‘ear’ of *Dactylogyrus pterocleidus* as well as *Chauhanellus auricalatum* (Figure 3.2 H).

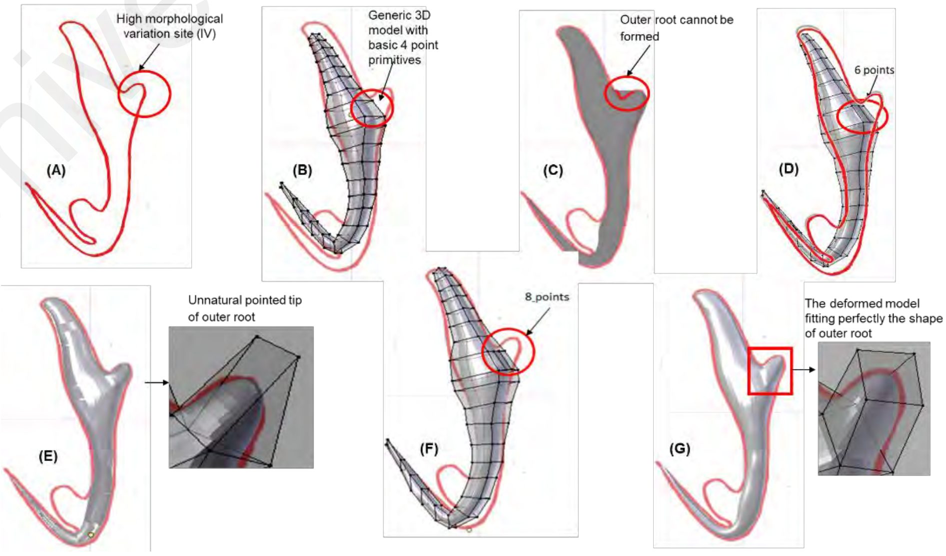


Figure 3.7: Process to determine the minimum set of point primitives in sites of high morphological variations.

Table 3.1: Number of point primitive on each site of high morphological variation.

Sites of morphological variation	**Number of point primitives**
Sites without variations (Original)	4
I	8
II	8
III	8
IV	8
V	8
VI	16
VII	16

3.2.6 Formation of final generic 3D model

The Cartesian coordinates of these additional point primitives were fed into the previous Python script of the preliminary 3D model to generate the final deformable generic model (Figure 3.8).

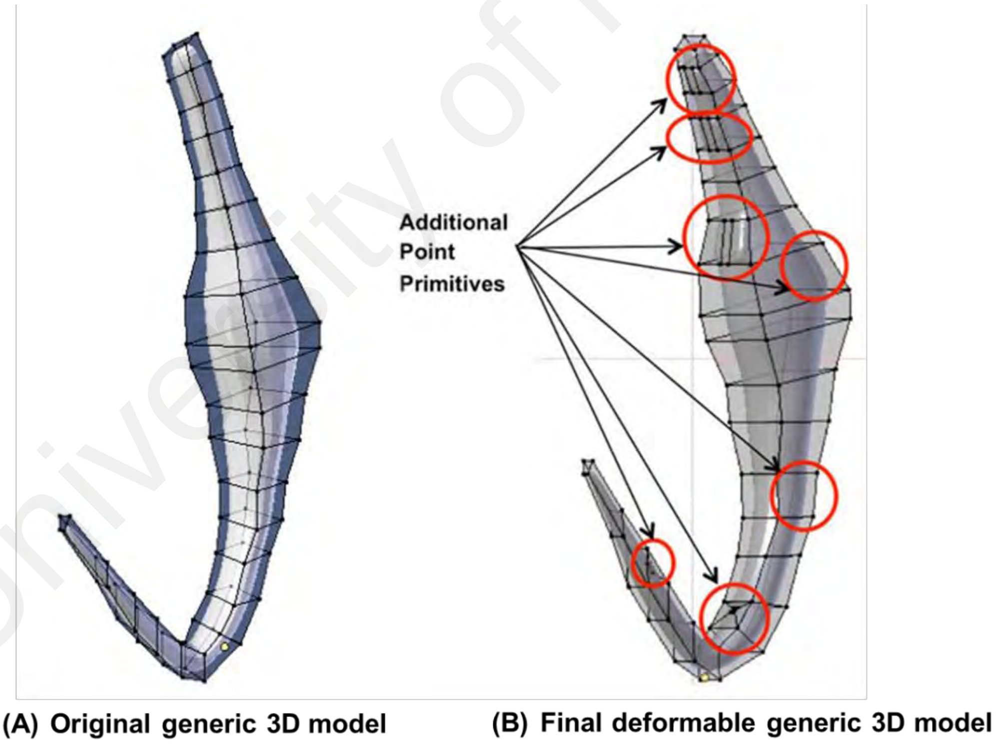


Figure 3.8: Comparison between original generic 3D model with final deformation generic model.

3.3 Phase 2: Data preparation for machine learning

In this study, the training data used for machine learning is 2D illustrations of monogenean anchor. A well-trained machine learning model requires a massive training data which can range from hundreds to millions to minimize the possibility of learning misleading or irrelevant patterns found in the training data (Chollet, 2017). However, one main issue encountered when preparing data for machine learning is the limited availability of 2D illustrations of monogenean anchor which could be extracted from existing publications. It is also tedious to manually extract the 2D illustrations by looking up publications one after another as training set. Besides, a more systematic storage system is also needed to store a massive dataset for machine learning. In this Phase 2, data augmentation was conducted by using a semi-automated approach that can generate thousands of synthetic illustrations as training set from merely a few original 2D illustrations of monogenean anchor. A database was then developed to store and manage those massive training set.

3.3.1 Data augmentation with a semi-automated approach

The core idea of this data augmentation approach is originated from a monogenean anchor morphometry study (Tan et al., 2010). In the morphometry study, Tan et al. (2010) have shown the variation of shape and size of a monogenean anchor can be found not only in interspecies level, but also in intraspecies level. The intraspecies variation means the shape of an anchor can vary from one specimen to another even if those specimens in study are belonging to a same species. In the work presented by Tan et al. (2010), the anchor shape variation in both intraspecies and interspecies level were analyzed based on five parameters: inner root (IR), outer root (OR), inner length (IL), outer length (OL), and point (pt) as shown in Figure 3.9.

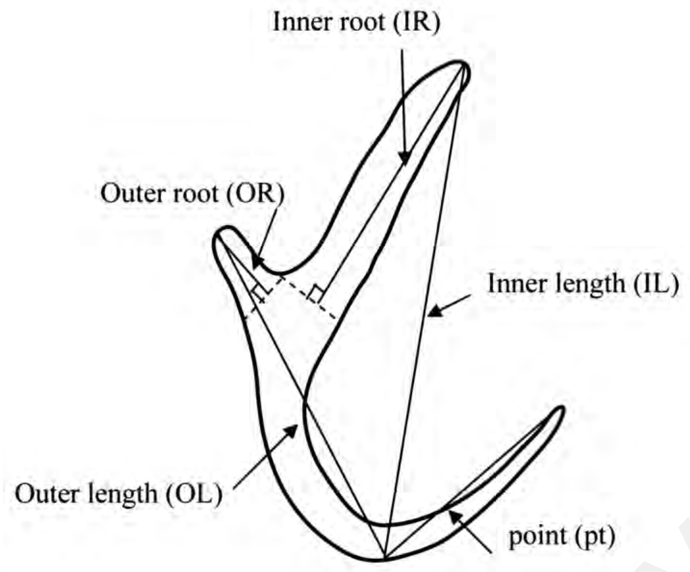


Figure 3.9: Anchor of *Trianchoratus* species. Adapted from Tan et al. (2010).

The aforementioned morphometry study has shown that a monogenean anchor of a species could possibly exist in a specific pattern of shape variation with different magnitude of IR, OR, IL, OL and pt. Inspired by this shape variation pattern shown in the morphometry study, a semi-automated data augmentation approach was developed to generate a sequence of synthetic 2D illustrations of monogenean anchors with different length values of IR, OR, IL, OL and pt from an original 2D illustration (Figure 3.10).

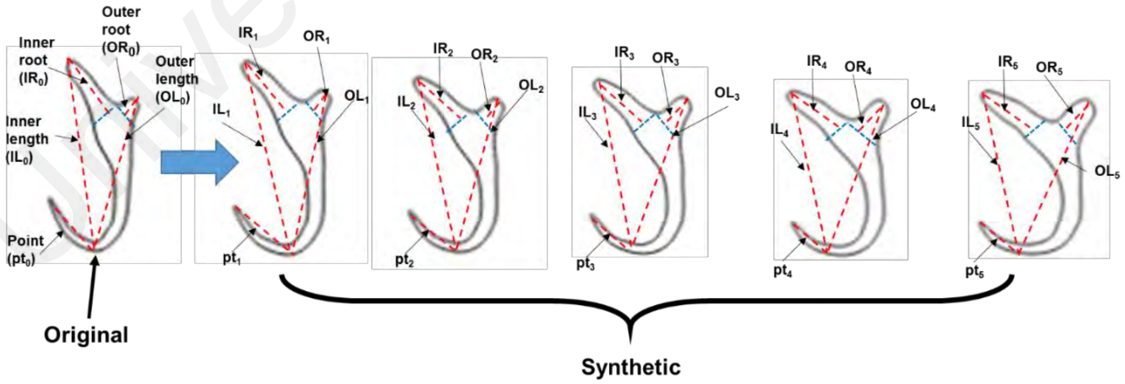
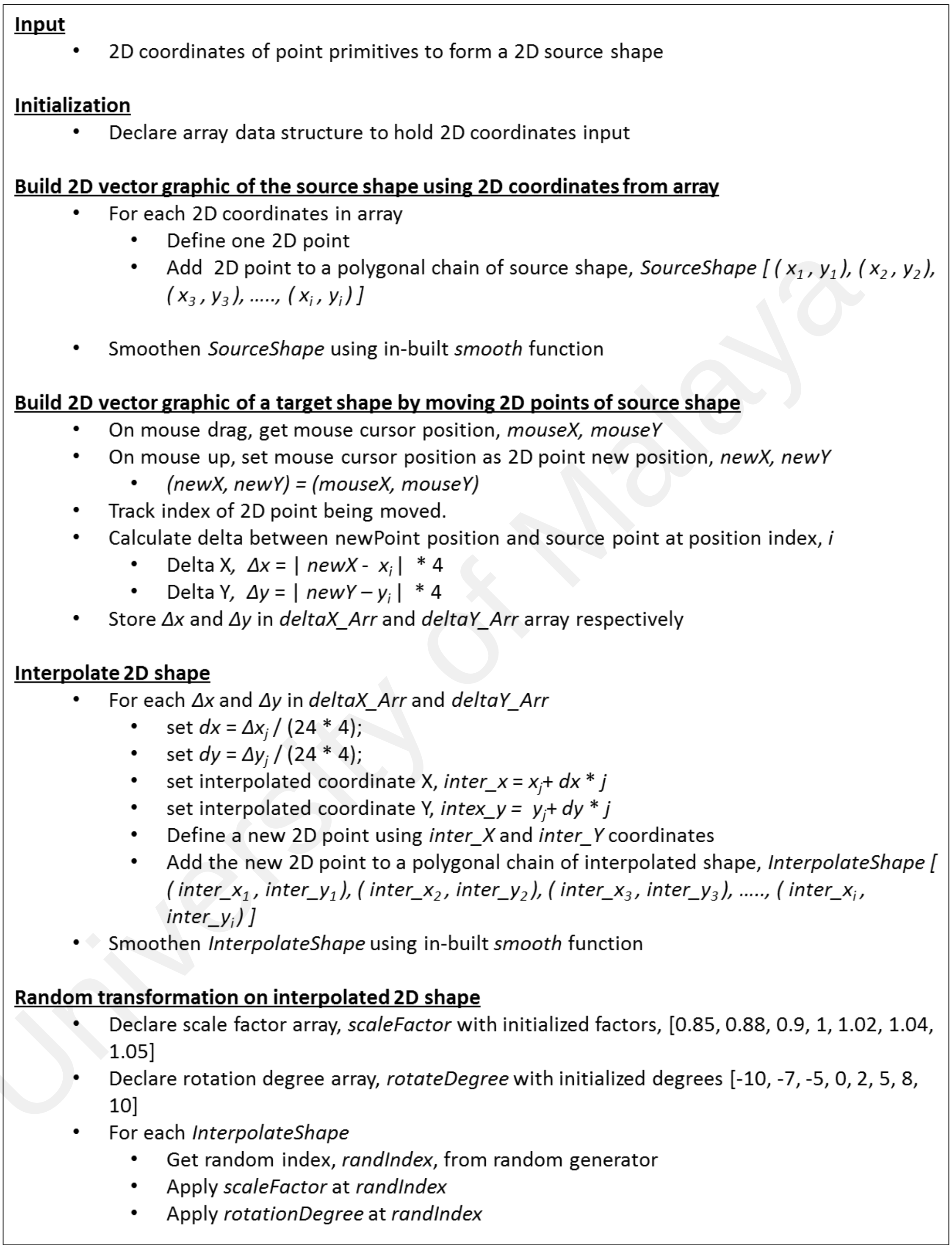


Figure 3.10: The data augmentation approach to generate the synthetic illustrations.

At the core of this semi-automated data augmentation approach is a 2D shape interpolation algorithm developed to generate a sequence of interpolated shapes of the anchor as the synthetic illustrations with different length values of IR, OR, IL, OL and pt from a process of morphing a source shape to a user-defined target shape. The logical details of how the 2D shape interpolation algorithm can morph a source shape to a user-defined target shape and generate the synthetic illustrations are described as follows:

I. An original illustration of monogenean is selected as the source shape and a 2D vector graphic of the source shape is generated. 2D vector graphic is a computer graphic structure which is composed of a sequence of point primitives. These point primitives are joined to form a 2D shape and each of the point primitive can be flexibly deformed by moving it around in a Cartesian coordinate space to form a new 2D vector graphic.
II. The point primitives of the source shape are moved to form a new 2D graphic vector which will become the user-defined target shape. In the process, whenever a point primitive of the 2D source shape is moved, the latest position of that point in the 2D Cartesian space is recorded.
III. The delta values between the original 2D coordinates of the source shape and the user-defined target shape are calculated. The delta values are broken down into a series of scale factors which are used to multiply with the original 2D coordinates to generate interpolated shapes iteratively from the source shape.
IV. A random image transformation is applied on the generated interpolated 2D shape by scaling and rotating them in different random degrees. This random image transformation is important to produce the interpolated 2D shapes with different orientations and sizes to cater the need of producing a more generalized synthetic training set to minimize the chance of overfitting a machine learning model.

A pseudocode was drafted (Figure 3.11) based on the 2D shape interpolation algorithm presented above and was used as a blueprint to develop a data augmentation program using Hypertext Markup Language (HTML) and JavaScript with inclusion of a specific 2D vector graphic library (Paper.js).



```
Input
    • 2D coordinates of point primitives to form a 2D source shape

Initialization
    • Declare array data structure to hold 2D coordinates input

Build 2D vector graphic of the source shape using 2D coordinates from array
    • For each 2D coordinates in array
        • Define one 2D point
        • Add 2D point to a polygonal chain of source shape, SourceShape [ ($x_1$ , $y_1$), ($x_2$ , $y_2$),
          ($x_3$ , $y_3$), ….., ($x_i$ , $y_i$) ]

    • Smoothen SourceShape using in-built smooth function

Build 2D vector graphic of a target shape by moving 2D points of source shape
    • On mouse drag, get mouse cursor position, mouseX, mouseY
    • On mouse up, set mouse cursor position as 2D point new position, newX, newY
        • (newX, newY) = (mouseX, mouseY)
    • Track index of 2D point being moved.
    • Calculate delta between newPoint position and source point at position index, i
        • Delta X, $\Delta x = | newX - x_i | * 4$
        • Delta Y, $\Delta y = | newY - y_i | * 4$
    • Store Δx and Δy in deltaX_Arr and deltaY_Arr array respectively

Interpolate 2D shape
    • For each Δx and Δy in deltaX_Arr and deltaY_Arr
        • set $dx = \Delta x_j / (24 * 4)$;
        • set $dy = \Delta y_j / (24 * 4)$;
        • set interpolated coordinate X, $inter\_x = x_j + dx * j$
        • set interpolated coordinate Y, $intex\_y = y_j + dy * j$
        • Define a new 2D point using inter_X and inter_Y coordinates
        • Add the new 2D point to a polygonal chain of interpolated shape, InterpolateShape [
          ($inter\_x_1$ , $inter\_y_1$), ($inter\_x_2$ , $inter\_y_2$), ($inter\_x_3$ , $inter\_y_3$), ….., ($inter\_x_i$ ,
          $inter\_y_i$) ]
    • Smoothen InterpolateShape using in-built smooth function

Random transformation on interpolated 2D shape
    • Declare scale factor array, scaleFactor with initialized factors, [0.85, 0.88, 0.9, 1, 1.02, 1.04,
      1.05]
    • Declare rotation degree array, rotateDegree with initialized degrees [-10, -7, -5, 0, 2, 5, 8,
      10]
    • For each InterpolateShape
        • Get random index, randIndex, from random generator
        • Apply scaleFactor at randIndex
        • Apply rotationDegree at randIndex
```

Figure 3.11: Pseudocode for 2D shape interpolation algorithm.

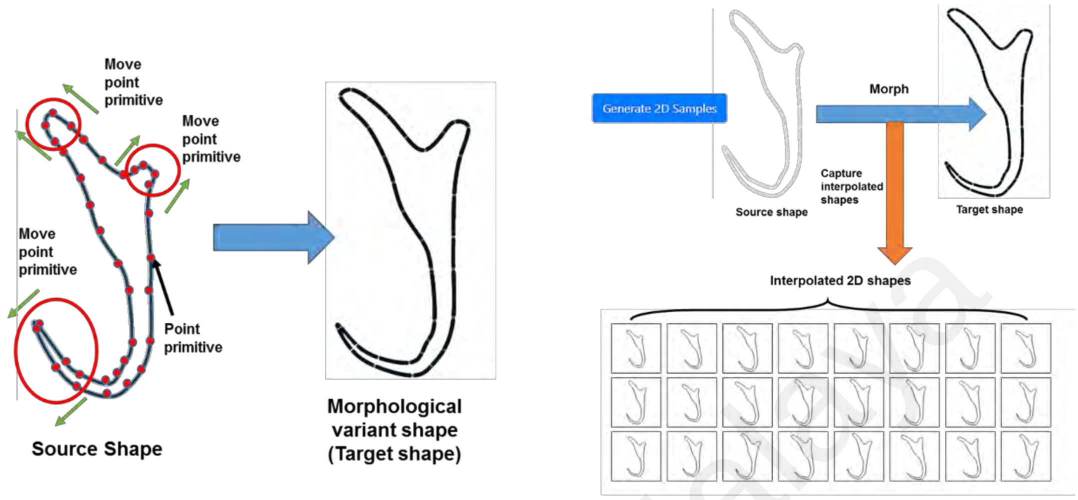
The data augmentation program was designed to generate 24 interpolated 2D shapes from one 2D source shape each time. 24 interpolated shapes were determined through a trial and error process to ensure the variation of the interpolated 2D shapes is not too subtle from one another and at the same time it enables production of a large amount of synthetic training data in a short time. Prior to using the data augmentation program, a 2D illustration of monogenean anchor extracted from a selected publication was used as the source shape. The selected anchor was uploaded onto Adobe Photoshop, followed by assigning point primitives along the anchor outline to extract the 2D coordinates for each point primitive (a process similar to Section 3.2.2). The 2D coordinates of the source shape were then fetched into the data augmentation program to generate a 2D vector graphic of the source shape. Once the source shape was generated in the data augmentation program, the following steps were done to augment the training set:

I. The 2D point primitives on the outer root, inner root and point regions of the source shape were selected individually and moved to a new position in the 2D Cartesian coordinate space to obtain a morphological variant shape which will be used as a target shape (Figure 3.12A).

II. The button "Generate 2D Samples" was activated to morph the source shape to the target shape obtained from the Step I. While the process of morphing was on-going, a total of 24 interpolated 2D shapes were captured and displayed on the grid cells (Figure 3.12B). Another enlarged cropped view of the 24 interpolated 2D shapes generated from the source shape is presented in Figure 3.13.

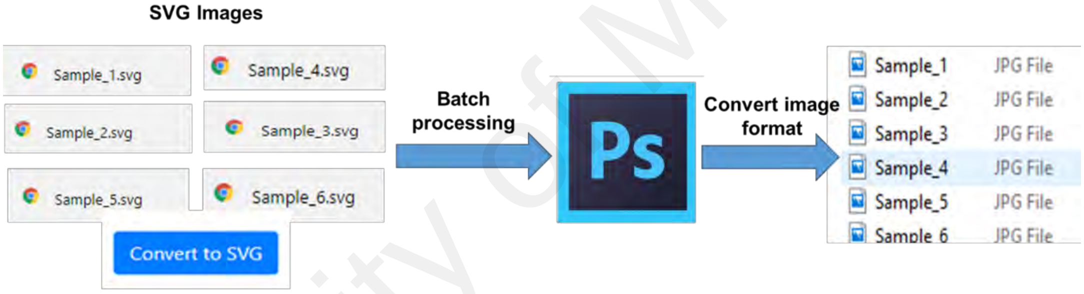
III. Next, the button "Convert to SVG" was activated to export the captured interpolated 2D shapes to SVG images. The SVG images obtained were then imported into Adobe Photoshop to undergo a batch processing to convert them into JPEG format (Figure 3.12C).

IV. Finally, the button "Record to Database" button was activated to store the pixel values along with the associated 2D coordinates of the constituent point primitives of each synthetic 2D illustration into a database (Figure 3.12 D). This storage process will be elaborated further in following Section 3.3.2.

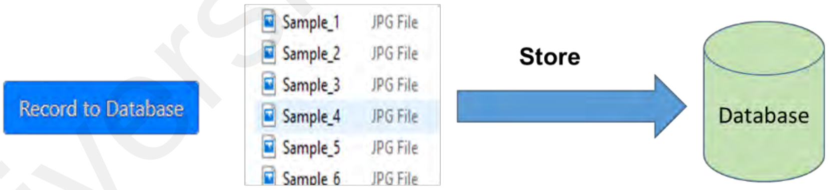


(A) Deform a source shape to a target shape

(B) Morph source shape to target shape



(C) Export 24 interpolated shapes to SVG images



(D) Store pixel values and associated 2D coordinates

Figure 3.12: Workflow to generate synthetic 2D illustrations using data augmentation program.

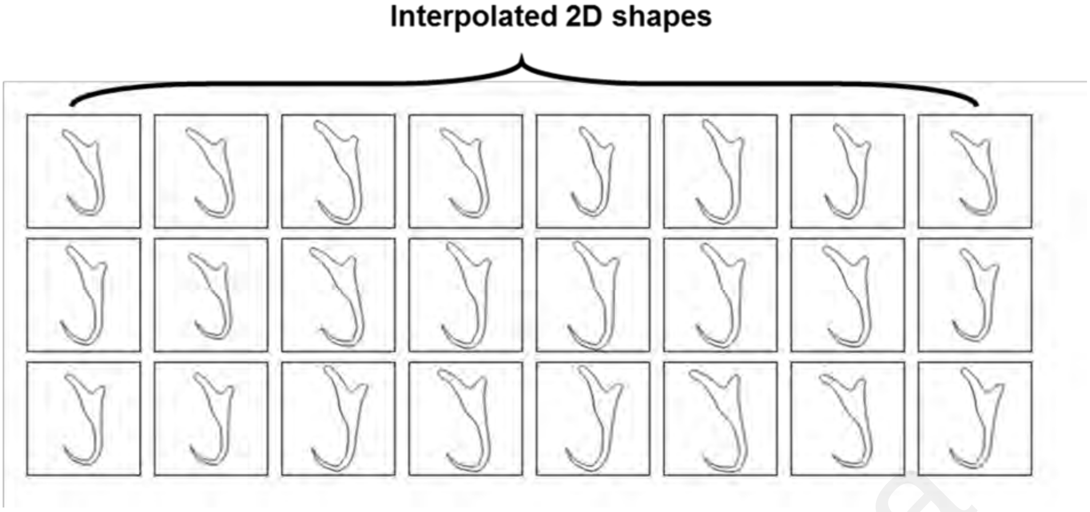
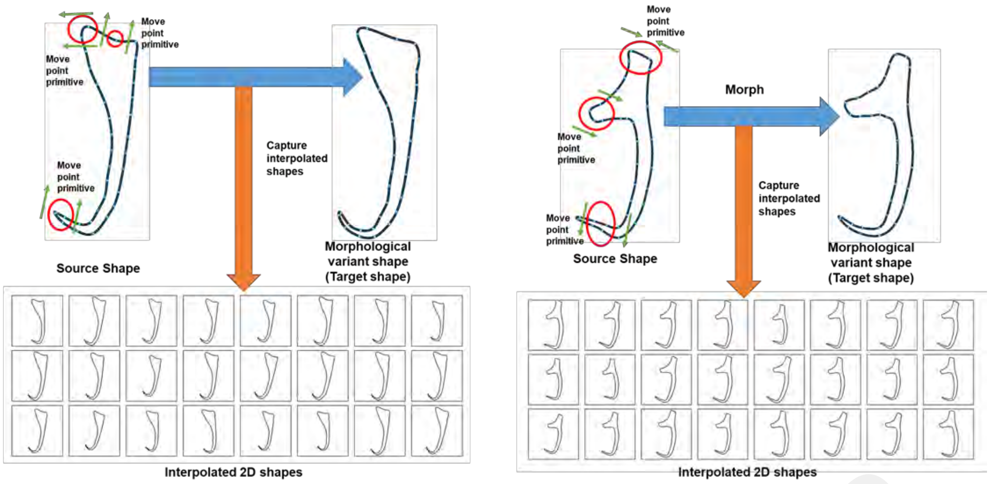


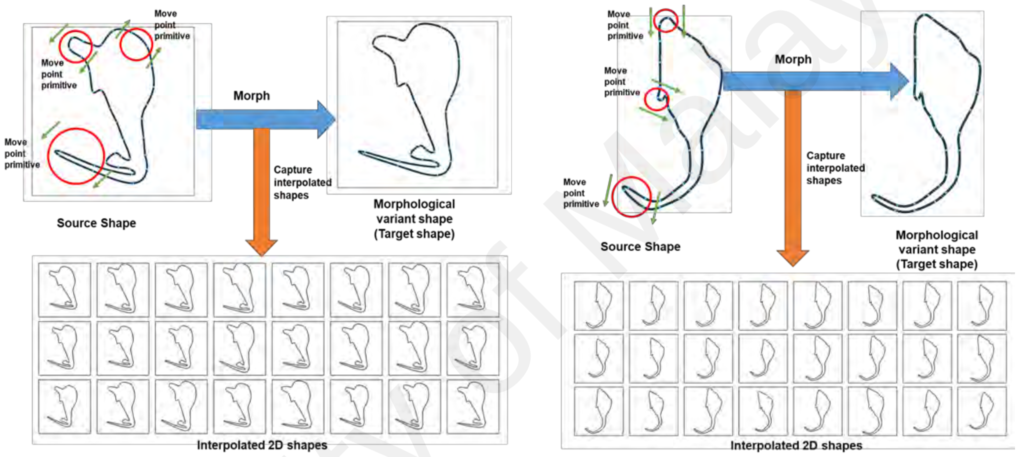
Figure 3.13: 24 interpolated 2D shapes.

The data augmentation procedure described above were repeated to generate 5000 synthetic 2D illustrations which consists of eight different categories with four of the categories share the shape features of noticeable inner root and outer root and another four categories without noticeable inner root and outer root (Figure 3.14). This is to cater the need for training a machine learning model which is versatile enough to detect the landmark points on all types of target monogenean anchors in this study. The number of generated synthetic illustrations had to be capped at 5000 due to the limited storage capacity in the local development platform. Besides, the generated synthetic 2D illustrations were verified by a monogenean expert through visual inspection.



(A) Generation of synthetic 2D illustrations for shape Category 5

(B) Generation of synthetic 2D illustrations for shape Category 6



(C) Generation of synthetic 2D illustrations for shape Category 7

(D) Generation of synthetic 2D illustrations for shape Category 8

Figure 3.14: Generation of synthetic 2D illustrations for shape categories without noticeable inner root and outer root.

The eight different categories of anchor shapes and their shape features are tabulated in Table 3.2 below.

Table 3.2: Eight categories of anchor shapes generated in augmented training sets.

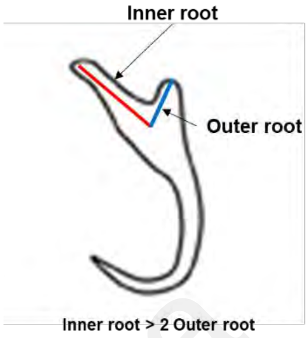
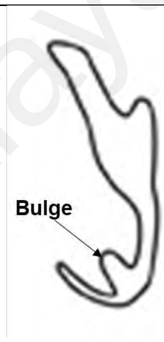
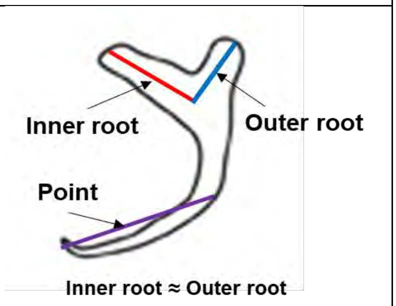
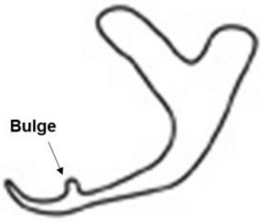
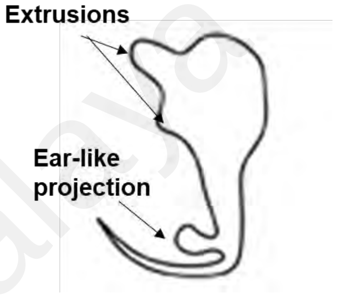
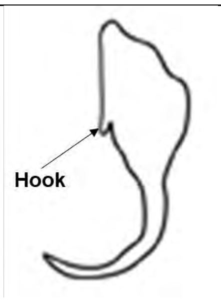
Shape category	Shape features	Sample illustrations
1	• y-shape • Inner root is at least more than twice of the outer root length	
2	• Shape similar to Category 1 except a bulge structure found nearby the bottom point of the anchor	
3	• y-shape • Inner root is approximately equal or slightly longer than the outer root • Width of inner root and outer root are relatively broader than Category 1 & 2 • Long length of point	
4	• Shape similar to Category 3 except a bulge structure found nearby the bottom point of the anchor	
5	• The structures of inner root and outer root are not noticeable	

Table 3.2, continued.

Shape category	Shape features	Sample illustrations
6	• Hammer-shape	
7	• Spherical head structure • Two extrusions found on the head of anchor • Ear-like projection nearby the bottom point	
8	• Spherical head structure • Hook like structure found on the head of anchor	

3.3.2 Database development

Prior to developing a database to store training sets, several considerations were made to choose the most suitable database system for this study. The considerations are as follows:

I. Database scalability

Database scalability refers to the ability of a database to allow it to store increasing amount of data without compromising its performance (Isaacson, 2014). The training set used for the machine learning in this study is expected to grow from time to time to obtain a more optimum learning model. Hence, a highly scalable database is needed to store a growing dataset and at the same time still maintains a reasonable performance.

II. Transaction speed

- Transaction speed refers to how fast the performance of a logical unit of work such as data retrieval or update in a database management system. The transaction speed of a database will directly affect the performance of an entire machine learning process. This is because the data retrieval is an essential step to fetch data from the storage system for training a machine learning model. A database system with high transaction speed is desirable.

III. Flexibility of data

- A flexible data model enables complex data to be stored efficiently (Dhillon, 2018). The storage data in this study includes the synthetic 2D illustrations which are represented as a huge array of pixel values and also the 2D coordinates of the point primitives that constitute every synthetic 2D illustration. A more flexible data model is needed to store this complex data.

IV. Cost of development

- The low implementation and infrastructure costs to maintain a database is part of the important factor that enables a long-term development of a research with limited funding resource. An open source database system is preferable for this study as it can run on inexpensive commodity hardware architectures.

V. Maturity

- A mature database refers to a storage system which has been well-established and is supported by wide community. A well-supported database system is more stable and provides database developer with API interfaces to build a computer program that can interact with the database system based on user specific requirements.

Based on the several considerations as aforementioned, a MongoDB database was developed to store the training set in this study. MongoDB is a NoSQL database which is free and open source. It is well-established with many API interfaces available in various programming languages such as Python, Java, C++, C# and etc and it is also supported by wide community. Compared with traditional relational database such as MySQL where the data model is based on a rigid schema, MongoDB's data model is document oriented and it has a flexible data structure that consists of a set of property names along with their associated values structured in JSON format. The flexibility of data model in MongoDB enable a database developer to represent data without conforming to a predefined schema. Besides, MongoDB database is also known to have much faster transaction speed compared with other NoSQL databases (e.g. RavenDB, CouchDB, Cassandra etc) and relational databases (Li & Manoharan, 2013).

The development of the MongoDB database for this study was initiated by defining a data model (Figure 3.16). The data model is composed of a collection of documents which are "2D illustration document", "Pixels document" and "Landmarks document". In the MongoDB context, a collection is a container for structurally or conceptually similar documents. A document is a basic unit of data in MongoDB which encapsulates a group of related properties along with their associated values in a JSON object. In the data model, the "2D illustration document" was defined as a parent node to hold all the relevant 2D illustration information in a JSON object, which consists of four properties which are "name", "shapeID", "pixels" and "landmarks". The "name" property was defined to hold an illustration identifier with maximum 100 characters whereas the "shapeID" is a numerical type property to hold the type of 2D shape based on the shape features categorized in Table 3.2. The "pixels" and "landmarks" properties were defined to hold the "Pixels document" and "Landmarks document" respectively. The "Pixels document" and "Landmark document" can be considered as two child nodes associated with the parent node, "2D illustration document". In the child node "Pixels document", there is only one property defined to hold an integer array of pixel values whereas there are two properties defined in "Landmark document" to hold an array of 2D coordinates, "coordinate_X" and "coordinate_Y" of each constituent point primitives that form a synthetic illustration.

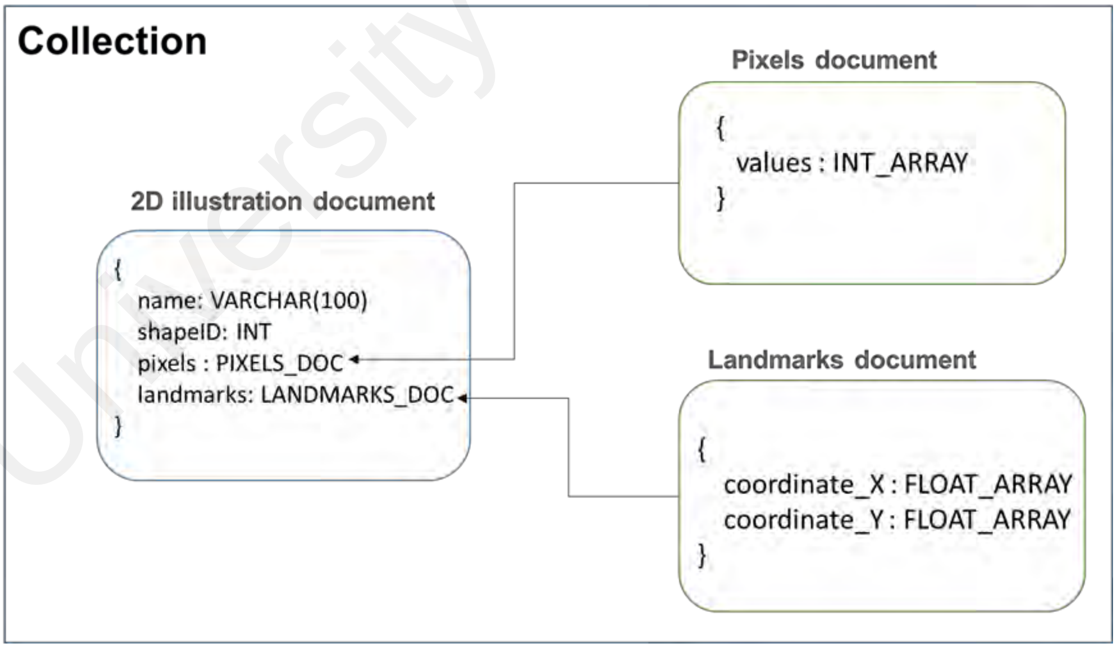


Figure 3.15: Data structure for MongoDB database model.

Based on the data model described above, a MongoDB database was created by writing a Python script, followed by creating an algorithm that could fetch data from the data augmentation program and store it in the MongoDB database. The logical details on how the algorithm was designed to store the synthetic illustration data in the database are described as follows:

I. The synthetic illustrations in JPEG format which were obtained from the data augmentation step (See Section 3.3.1) are converted to grayscale images.

II. The grayscale images are resized into 96 x 96 pixels values. The pixel values for each synthetic illustrations are then stored in an array. The dimension of each image is scaled to 96 x 96 pixels values because this is one of the common image resolution chosen for machine learning studies for image analysis (Coates et al., 2011; Khorrami et al., 2015; Zhong et al., 2012; Zhu et al., 2013).

III. On the other hand, the 2D coordinates of the constituent point primitives for each synthetic 2D illustration are stored in another array.

IV. The two arrays containing the pixel values and the 2D coordinates of the constituent point primitives along with the relevant metadata (image name and shape ID) are then fetched into the MongoDB database.

A pseudocode that follows the logical details of the data storage process as presented above is shown in Figure 3.16. This pseudocode was then used as a blue print to develop a Python module to store all the synthetic illustrations obtained from the 2D augmentation program into the MongoDB database. This Python module was written using Python based MongoDB API and OpenCV API to handle the database operation tasks (e.g. insert, update & delete data) and image processing tasks (e.g. converting to grayscale & resizing image) respectively. This Python module was integrated with the data augmentation program and it can be executed by activating the “Record to Database” button in the data augmentation program. The reason of such integration is to streamline the process of data augmentation and data storage in one single and consistent user interface which is timesaving, especially to prepare a large volume of training sets for the machine learning in later stage.

```
Input
    • Synthetic 2D illustrations in JPEG format
    • 2D coordinates of point primitive for each synthetic 2D illustration

Initialization
    • Declare two arrays to keep pixel values and 2D coordinates
    • Declare another two variables to keep illustration identifier name and
      shape ID

Image Processing
    • Convert synthetic 2D illustration to grayscale using OpenCV API
    • Resize it to 96 x 96 pixels using OpenCV API

Set pixel values and 2D coordinates to arrays
    • Let pixel array, pixel_values = [255 255 225…245 246 237]
    • Let coordinate array, coord_2D = [[3.5, 4.5], [3.7, 4.7], [3.8, 4.9]….]

Fetch illustration info to database
    • Let illustration identifier name, illus_name = "Sample xx"
    • Let shape ID, shapeID = X
    • Save illus_name, shapeID, pixel_values, & coord_2D into database using
      MongoDB Python API.
```

Figure 3.16: Pseudocode of storing illustration data.

3.4 Phase 3: Machine learning

In this phase, an Artificial Neural Network (ANN) algorithm was used to build the machine learning model to enable landmark detection on an input illustration. While there are other alternative machine learning models such as Random Forest and Support Vector Machine which can be deployed for landmark detection, ANN was selected because it has shown great success in visual pattern recognition in recent years (Krizhevsky et al., 2012). Besides, a same ANN model could be optimized using more training set for scalability in long term development. In terms of technical aspect, it is also easy to access the latest and well-established ANN APIs offered by technology giants such as Google and Microsoft. In this study, an open source neural network API, Keras was used to build the ANN. Keras is a Python-based neural network API which is officially supported by Google and it offers high-level modular machine learning functions that can ease the neural network development with just several lines of Python scripts. A Python script was written to use the Keras API to build a deep neural network based on ANN architecture which consists of one input layer with a 2D tensor, two hidden layers with 512 nodes and 128 nodes respectively and one output layer with 68 numerical values (Figure 3.17). The 2D tensor in the input layer denotes a 2D vector that holds an array of 96 x 96 pixels values of every 2D illustration training set. In the hidden layers, the ReLU was used as the activation function to calculate the weighted sum of the input nodes. The 68 numerical values in the output layer are predicted 2D coordinates (x, y) for 34 point primitives. These predicted point primitives are regarded as the landmark points in the context of this study. The trained neural network is aimed to detect these landmark points on an input 2D illustration.

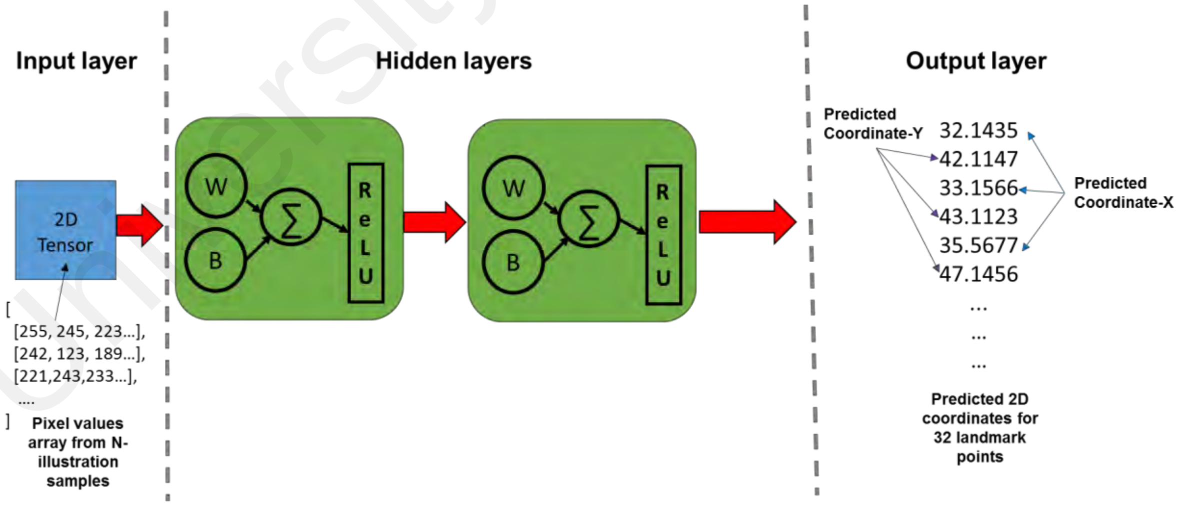


Figure 3.17: Supervised learning for pixel values mapping.

The ANN was trained via a supervised learning process by mapping the pixel values of all the synthetic illustrations retrieved from the MongoDB database to each of their associated 2D coordinates of constituent point primitives (Figure 3.18). Before training the ANN, all pixel values samples were divided into a training set (80% of samples) and a test set (20% of samples). The test set contains independent samples which are used to evaluate the accuracy of the trained model in a later stage. Besides, 30% of the samples were set apart from the training set to be used for model validation during the training process. This ratio of 70/30 (training sets/ validation sets) was adopted after a prior analytical test was conducted to repeatedly train the same pixel values samples with ratio of 80/20, 70/30, and 60/40. The ratio of 70/30 was found to give the highest test accuracy of landmark detection (See Chapter 4, Section 4.4).

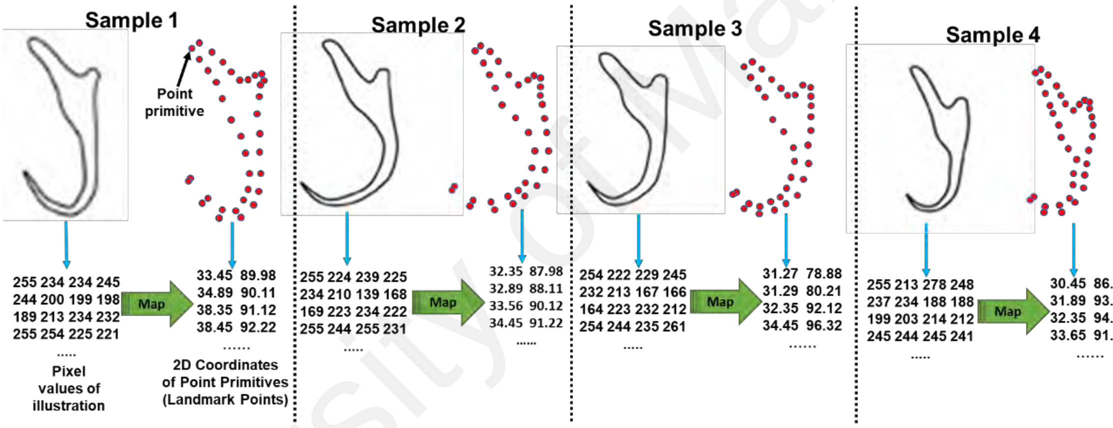


Figure 3.18: Examples of supervised learning to map pixel values of illustrations to each of their associated 2D coordinates of constituent point primitives.

The procedures of training the ANN are as follows:

I. Mean Square Error (MSE) was used as the loss function to measure how good the preliminary machine learning models are obtained during the training process to predict the expected outcome.

II. Next, Stochastic Gradient Descent (SGD) optimizer was applied to repeatedly update the weight parameters of the preliminary machine learning models to reduce the loss values during the training process. The hyperparameters selected to configure the optimizer are as follows:

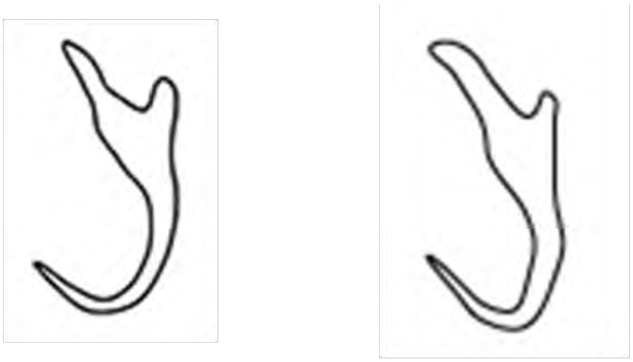
    a. Learning rate = 0.01
    b. Momentum = 0.9
    c. Decay rate = 0
    d. Nesterov momentum = True

III. The training process was done for 200 epochs (200 iteration over all training samples) in mini batches of 20 samples. During the training process, the ANN model was tested with the validation sets in every epoch.

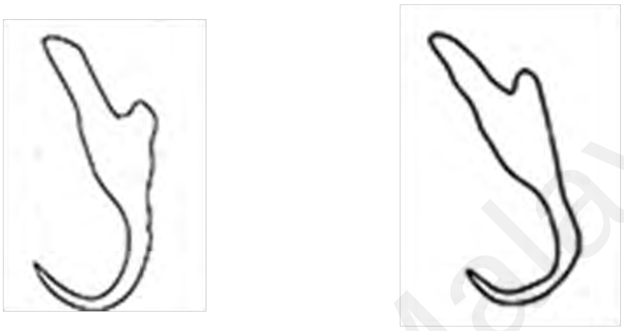
IV. Upon completing the training process, the trained model was evaluated using the test set.

V. At last, the trained model was exported to a H5 file.

The trained ANN was first tested on the monogenean anchors of the four selected species (Figure 3.19) which share the common shape feature of Category 1 (Refer Table 3.2). It was then followed by testing the trained ANN on another eight selected monogenean species which possess a disparity pattern of shapes (Figure 3.20). This is aimed to test the versatility of the trained machine learning model to detect landmark points on the anchors which share the similar shapes as well as with very different shape from each other. All the 12 illustration samples of monogenean anchors were extracted from publications (Gusev, 1976; Rahmouni et al., 2017; Řehulková et al., 2013; Steenberge et al., 2015).

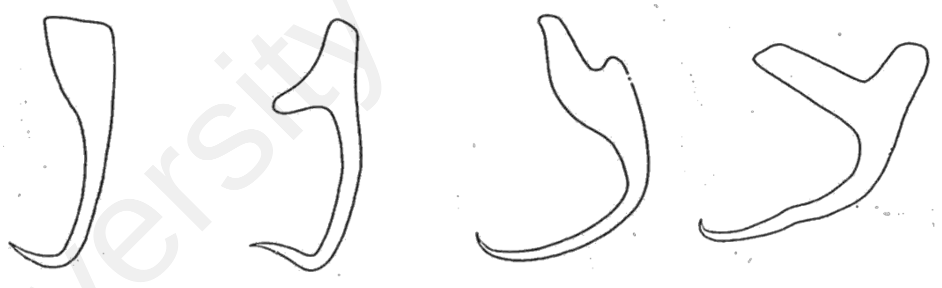


(A) *Cichlidogyrus dracolemma* (B) *Dactylogyrus wunderi*

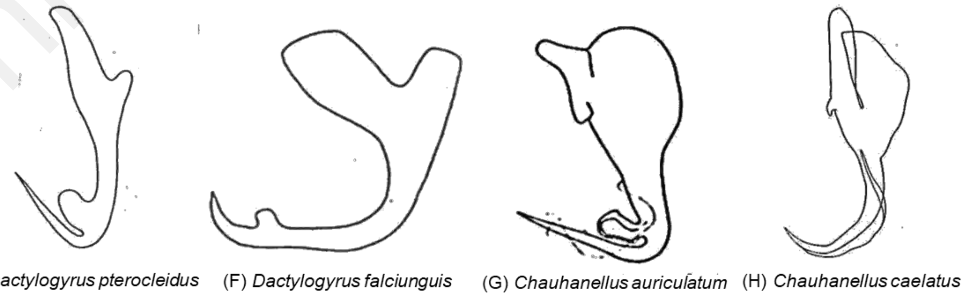


(C) *Cichlidogyrus raeymaekersi* (D) *Cichlidogyrus aspiralis*

Figure 3.19: 2D illustrations of four selected monogenean anchors from Gusev (1973), Steenberge et al. (2015), Rahmouni et al. (2017) and Řehulková et al. (2013).



(A) *Dactylogyrus primarius* (B) *Dactylogyrus primarius* (C) *Dactylogyrus falcatus* (D) *Dactylogyrus vastator*



(E) *Dactylogyrus pterocleidus* (F) *Dactylogyrus falciunguis* (G) *Chauhanellus auriculatum* (H) *Chauhanellus caelatus*

Figure 3.20: 2D illustrations of eight selected monogenean anchors from Gusev (1973) and Lim (1994).

3.5 Phase 4: Formation of automated 3D modelling pipeline

In this phase, a 3D modelling system which is composed of three modules was developed. The three modules are dedicated to input processing, landmark detection and deformation of generic 3D model respectively. In addition, the machine learning model obtained from Phase 3 was also integrated into the 3D modelling system. The three modules along with the integrated machine learning model work synergistically to form an automated 3D modelling pipeline which can automatically deform the digital generic 3D model obtained from Phase 1 into a target 3D shape by using a 2D illustration as an input (Figure 3.21).

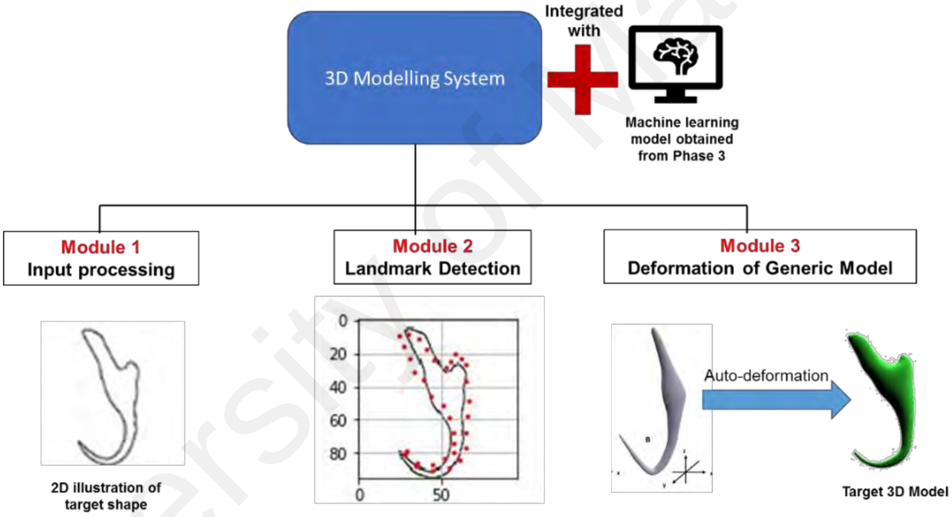


Figure 3.21: 3D modelling system design.

The logical details on how the three modules were designed to form the automated 3D modelling pipeline are as follows:

A. Module 1: Input processing (Figure 3.22A)

The input of this automated modelling pipeline is a 2D illustration of a target shape and this module is supposed to read and process the input illustration. The input illustration is translated into an array of pixel values in an interval of 0 to

255. These pixel values are normalized by dividing each of the pixel by 255 to re-scale the pixel values in an interval of 0 to 1. The normalization of the pixel values is important to standardize the scale of the pixel values which will be fetched into a machine learning model to obtain a predicted output.

B. Module 2: Landmark detection (Figure 3.22B)

This module is to detect 34 landmark points on the input illustration. This is achieved by loading the machine learning model (a h5 file) obtained from Phase 3, followed by fetching the normalized pixels into the model to predict 2D coordinates for each landmark point. The predicted output is a 1D array of 68 numerical values that signify coordinate-X and coordinate-Y for the 34 landmark points. The 1D array of predicted coordinates are reshaped into a 2D array with dimension of 34 x 2. Each element in the 2D array is a 1D array of coordinate-X and coordinate-Y for a landmark point.

C. Module 3: Deformation of generic 3D model (Figure 3.22C)

This module is to deform the digital generic 3D model obtained from Phase 1 into the target shape. To achieve this aim, the generic 3D model is loaded into the system. Each of the constituent point primitives of the generic model are aligned with a corresponding landmark point signified by the predicted coordinate values obtained from module 2. The alignment between the original point primitives of the generic 3D model with the predicted landmark points causes the deformation of the generic model into the target shape.

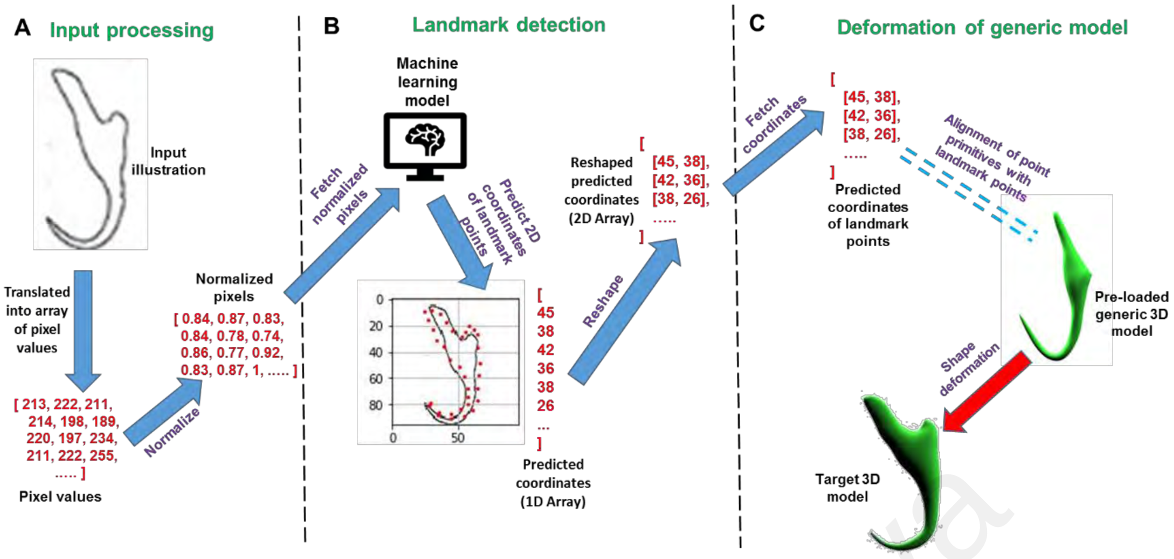


Figure 3.22: Automated 3D modelling pipeline.

A pseudocode which features the operating logic of the automated 3D modelling pipeline was drafted (Figure 3.23) and this pseudocode was used as a blueprint to develop the 3D modelling system by using HTML, JavaScript and Python programming languages.

```
Module 1: Receiving and processing Input
    • Declare variable, img, to hold pixel values of input illustration
    • Read image file as an array of pixel values and assign it to img variable
    • Normalize pixel values of input illustration, using formula:
        o set norm_img = img / 255
    • Transform normalize pixel values into 2D array
        o norm_img          2D Tensor with shape(1, 96 * 96)

Module 2: Landmark Localization
    • Load machine learning model (a h5 file) into system
    • Fetch 2D array of norm_img into machine learning model
        o model.predict(2D tensor norm_img)
    • Declare a variable, predicted_output, to hold the values of predicted landmark coordinates
        o Set predicted_output = model.predict(2D tensor norm_img)
    • Reshape 68 numerical output values into 34 pairs of coordinates and add them into a two-
      dimensional array, coordinate_array
        o For each 68 predicted_output values with index, i = 0, 1, 2, 3…67
              if index is even
                  Set coords_x = predicted_output
              if index is odd
                  Set coords_y = predicited_output

          coordinate_array.push(coords_x, coords_y)

Module 3: Deformation of generic model
    • Set a 3D scene
    • Load the generic 3D model from Phase 1 into 3D scene
    • Align coordinates of point primitives of generic 3D model, point_primitive_x &
      point_primitive_y, with coordinate_array obtained in previous step.
      (This step shall deform the generic 3D model)
        o For each coords_x , coords_y in coordinate_array with index, i = 0, 1, 2, 3…67
            o Set $point\_primitive\_x_i = coords\_x_i$
            o Set $point\_primitive\_y_i = coords\_y_i$

    • Smoothen deformed 3D model using Catmull_Clark modifier to obtain final 3D mesh
    • Add smoothened 3D model to 3D scene
```

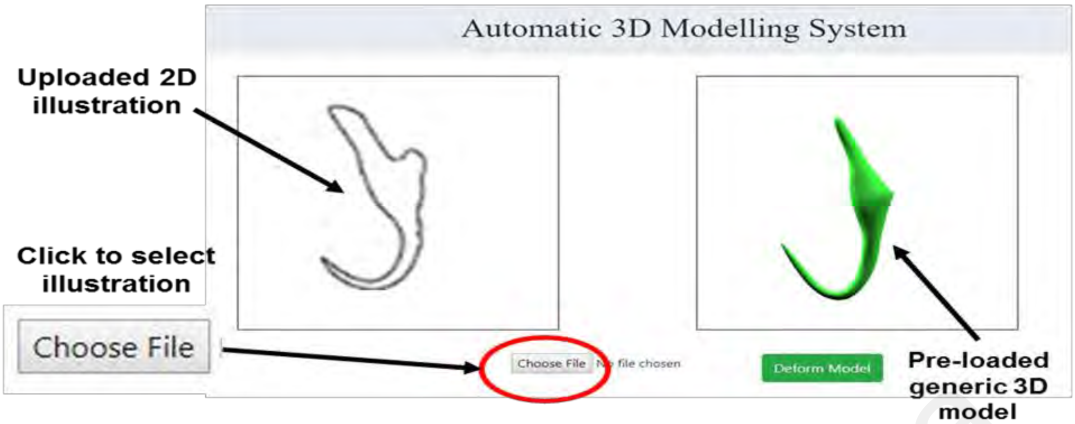
Figure 3.23: Pseudocode for the development of automatic 3D modelling system.

This 3D modelling system was built as a full-stack web application which consists of front-end and backend components. The front-end component includes a simple user interface created using HTML canvas element, which allows user to upload an input illustration as a target shape. This user interface also includes another canvas element which is pre-loaded with the generic 3D model obtained from Phase 1 and a user can interact with the 3D model by rotating it in 360 degrees for 3D visualization. On another hand, the backend components of the system are basically the logical parts that implement the operation logics of the three system modules as mentioned above to process the input illustration, to detect landmarks on the input illustration and to load and deform the generic 3D model and finally render it on to the interface. The development of this backend components was heavily dependent on a specialized JavaScript 3D library which is Three.js. Three.js is a modern 3D web technology that enables a developer to create a program to construct, manipulate and display a 3D model on a web browser without any additional plug-in 3D viewer program.

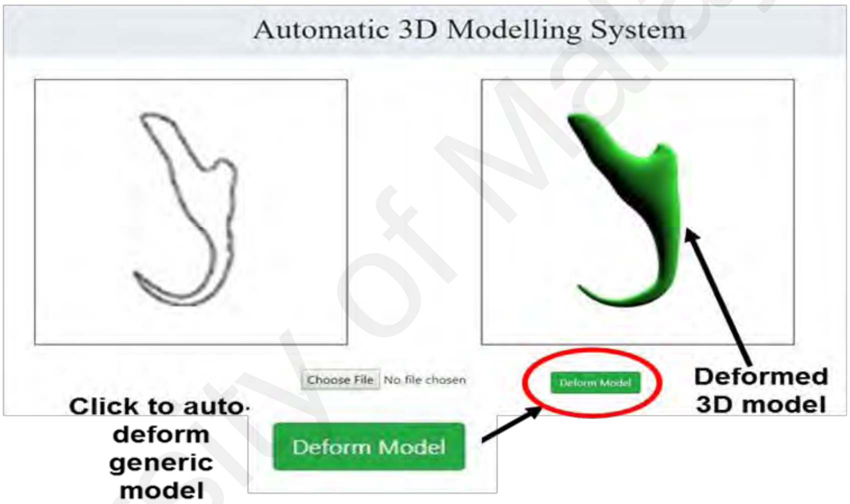
The procedures of using the developed 3D modelling system to implement the automated 3D modelling pipeline are as follows:

I. A 2D illustration of a selected monogenean anchor was uploaded onto the system interface. This was done by activating the "Choose File" button and followed with selecting a 2D illustration as the target shape. (Figure 3.24A)

II. Next, the button "Deform Model" was activated (Figure 3.24B) to fetch the input 2D illustration to the back-end components of this system to enable it to go through the entire 3D modelling pipeline (input processing, landmark detection and deformation of generic model). This entire modelling pipeline was completed by deforming the preloaded generic 3D model into the target 3D shape in an automated way without any human intervention.



(A) Upload 2D illustration



(B) Generate 3D model

Figure 3.24: Implementation of the automated 3D modelling pipeline.

The procedure of using the 3D modelling system as described above was repeated to generate the 12 selected monogenean anchors through the automated 3D modelling pipeline and the results are presented in Chapter 4.

3.6 Phase 5: Result evaluation

3.6.1 Manual deformation of the generic 3D model

To address the first question, a manual deformation approach was carried out to deform the digital generic 3D model obtained from Phase 1 into 8 different forms of 3D anchors. Each of them possess one of the eight categorized shape features as tabulated in Table 3.2. This is to examine whether the digital generic 3D model is deformable and is versatile enough to derive multiple types of the 3D anchor. The implementation of the manual deformation approach is described as follows:

I. The 2D illustrations of a target anchor is imported into Blender. It was followed with loading the generic 3D model onto the Blender and the generic model was aligned onto the 2D illustration (Figure 3.25A).

II. Next, the generic 3D model was deformed by first selecting one or more point primitives on the surface of the generic model (Figure 3.25B) and moving them to fit the outline of the 2D illustration (Figure 3.25C). The shape of the generic 3D model was deformed following the movement of selected point primitives.

III. The deformation was repeated by selecting other point primitives and moving them to fill up the 2D illustration (Figure 3.25D & Figure 3.25E) to obtain the desired 3D shape (Figure 3.25F).

IV. The Step I, II and III were repeated for all the 8 selected monogenean anchors which possess different shape feature ranging from Category 1 - 8 as tabulated in Table 3.2.

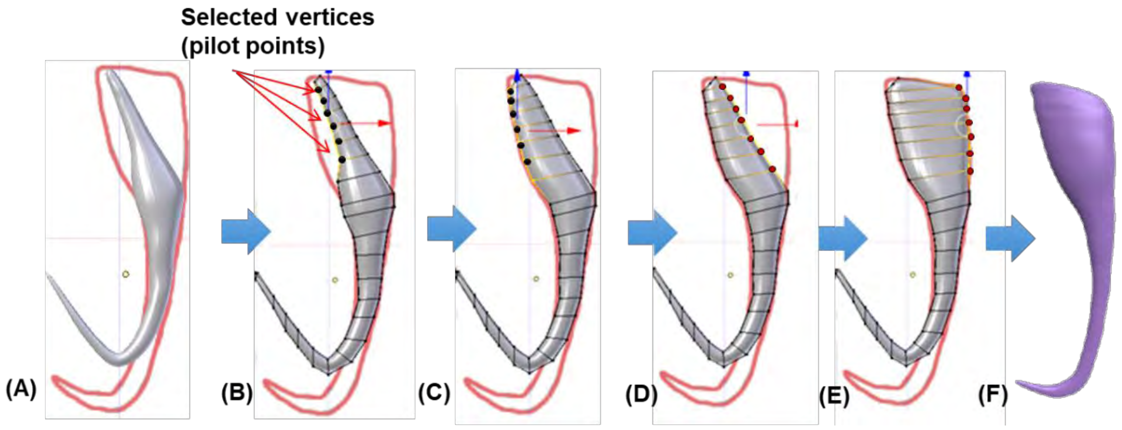


Figure 3.25: Process of deformation of generic 3D model into desired shape by manual deformation technique.

3.6.2 Evaluation of the automated 3D modelling pipeline

To address the second question, two evaluation approaches were carried out to examine the efficiency of the automated 3D modelling pipeline to construct all the target monogenean anchors by deforming the digital generic 3D model in an automated way. The first evaluation approach was to compare the 3D models of the monogenean anchors produced by both manual deformation method and the automated 3D modelling pipeline. This was done by a visual inspection on the two same groups of 3D anchors produced by the manual and automated deformation approaches to examine how well the resulting 3D models fit the outlines of their respective 2D templates.

The second evaluation approach was based on a more quantitative analysis which was done by calculating the Euclidean distance between the 34 predicted landmark points and the 34 control points assigned along the edges of the input illustrations of anchor. This evaluation is based on a postulate that the smaller the Euclidean distance between the predicted landmarks and their corresponding control points on the illustration, the shape accuracy of the resulting 3D model will be higher. The Euclidean distance serves as a fine and precise indicator to show the disparity pattern between the predicted landmark points and their corresponding control points (Figure 3.26).

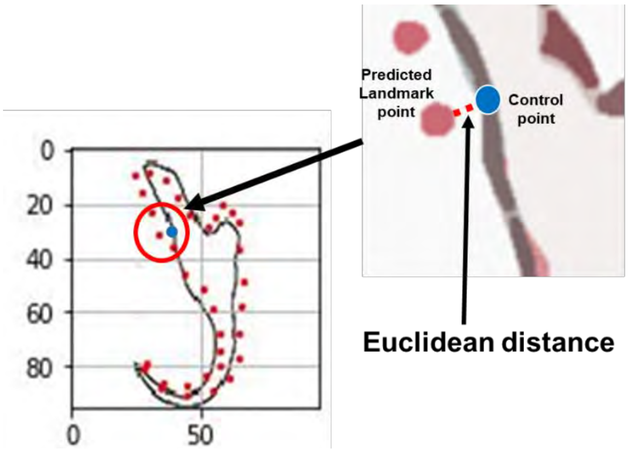


Figure 3.26: Euclidean distance between a predicted landmark point and a corresponding control point.

The steps of the quantitative analysis based on the calculation of Euclidean distance between the control points and the predicted landmark points on one input 2D illustration are as follows:

I. A sequence of control points were manually annotated on the ideal positions along the edge of an input 2D illustrations (Figure 3.27A). This was done by uploading one input 2D illustration to Adobe Photoshop and a total of 34 control points were manually annotated along the edge of the 2D illustration.

II. The coordinates of the control points were extracted from the Properties Window in Adobe Photoshop and were stored in a CSV file (Figure 3.27B).

III. The same input 2D illustration was fetched to the 3D modelling system and the landmark point coordinates predicted by the machine learning model were collected and stored in a CSV file (Fig. 3.27C). The collection of predicted landmark point coordinates was achieved by writing a Python script to include an additional module which is dedicated to that task in the 3D modelling system.

IV. The coordinate sets of control points and predicted landmark points obtained from Step I-III were used to calculate the Euclidean distance using the Pythagorean formula as follows:

$$\textbf{Euclidean distance} = \sqrt{(\boldsymbol{C_x} - \boldsymbol{P_X})^2 + (\boldsymbol{C_Y} - \boldsymbol{P_Y})^2}$$

C_x = Coordinate-X of Control point

C_y = Coordinate-Y of Control point

P_x = Coordinate-X of Predicted point

P_y = Coordinate-Y of Predicted point

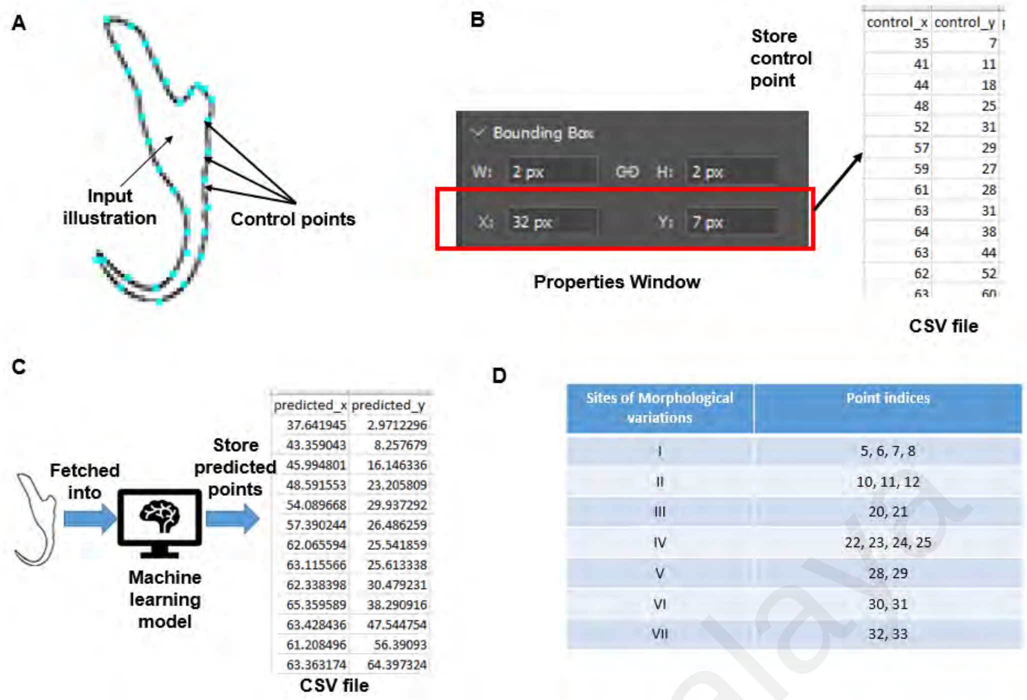
V. The Euclidean distances between 34 pairs of the control points were tabulated and each of them was indexed by a point index starting from 0 to 33. One point index refers to one pair of the comparative control point and predicted landmark point.

VI. The point indices were matched with one of the sites of high morphological variations (Figure 3.27D).

VII. Next, the tabulated Euclidean distances for all 34 pairs of the control points and predicted landmark points were plotted against the point indices in a line chart to show the degree of diversion of each predicted landmark point from their corresponding control point.

VIII. Each sites of high morphological variations were examined further to check for the existence of shape distortion in a derived 3D model. The observations on those sites and how they affect the resulting 3D model of all the 12 species are recorded and presented in Chapter 4.

IX. The Step I-VIII were repeated for all the 12 input 2D illustrations used for creating the target 3D anchors.



control_x	control_y
35	7
41	11
44	18
48	25
52	31
57	29
59	27
61	28
63	31
64	38
63	44
62	52
63	60

predicted_x	predicted_y
37.641945	2.9712296
43.359043	8.257679
45.994801	16.146336
48.591553	23.205809
54.089668	29.937292
57.390244	26.486259
62.065594	25.541859
63.115566	25.613338
62.338398	30.479231
65.359589	38.290916
63.428436	47.544754
61.208496	56.39093
63.363174	64.397324

Sites of Morphological variations	Point indices
I	5, 6, 7, 8
II	10, 11, 12
III	20, 21
IV	22, 23, 24, 25
V	28, 29
VI	30, 31
VII	32, 33

Figure 3.27: Procedures of collecting coordinate-X and coordinate-Y of control points and predicted landmark points.

CHAPTER 4: RESULTS

4.1 Introduction

In this chapter, the results obtained from the five phases of the project development are reported according to the sequence as follows:

A. Generic 3D models of monogenean anchor
B. Output of data preparation for machine learning
    I. Synthetic 2D illustrations
    II. MongoDB database
C. Landmark point detection on the input illustrations of monogenean anchors using the machine learning model
D. 3D models of monogenean anchors derived from deformation of the generic 3D model
E. Evaluation on the efficiency of the automated 3D modelling pipeline

4.2 Generic 3D model of monogenean anchor

In Chapter 3, a digital generic 3D model of monogenean anchor was created (Figure 4.1) as stated in Phase 1. This generic model was used as a 3D template for deriving 3D models using both the manual deformation approach and the automated 3D modelling pipeline. Although the structure of this digital generic 3D model is visually simple, it can be deformed into diverse form of anchor shapes (See Figure 4.24 and Figure 4.25).

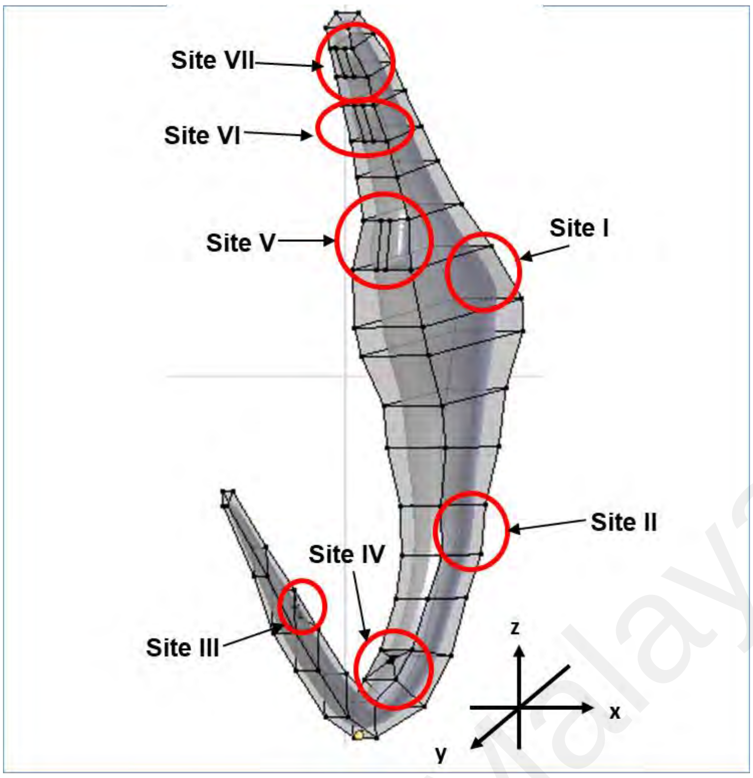


Figure 4.1: The digital generic 3D model developed in Phase 1.

4.3 Data preparation for machine learning

The results obtained from the Phase 2 (in Chapter 3) project development include augmented illustration training sets along with each of their annotated landmark point coordinates and a MongoDB database to store the training sets.

4.3.1 Synthetic 2D illustrations

A total of 5000 synthetic 2D illustrations of monogenean anchors were generated using the data augmentation program developed in this study as presented in the Chapter 3 (See Section 3.3.1). These 5000 synthetic illustrations of monogenean anchor were synthesized based on eight different categories of morphological variant shapes (Table. 3.2) to cater the need for training the machine learning model to detect the landmark points on the input illustrations of all the selected monogenean anchors. The breakdown number of the generated synthetic illustrations for each of the eight categories is given in Table 4.1.

Table 4.1: Number of synthetic illustrations generated for each category of anchor shape.

Category	**Number of generated synthetic illustrations**
1	1460
2	500
3	540
4	500
5	500
6	500
7	500
8	500

The most common type of anchor shape which was observed in the survey literatures is the first category. Hence, a much higher number of synthetic illustrations of anchor with the first category of shape were generated as training sets.

All the synthetic 2D illustrations are scaled and rotated in random degrees. This is to ensure a high degree of morphological variant shapes can be obtained to train a more generalized machine learning model. The shape variation of all the generated synthetic illustrations in each of the shape categories can be reflected by showing the distribution of the 34 constituent point primitives of every synthetic 2D illustrations in a 2D Euclidean space (Figure. 4.2 – 4.9).

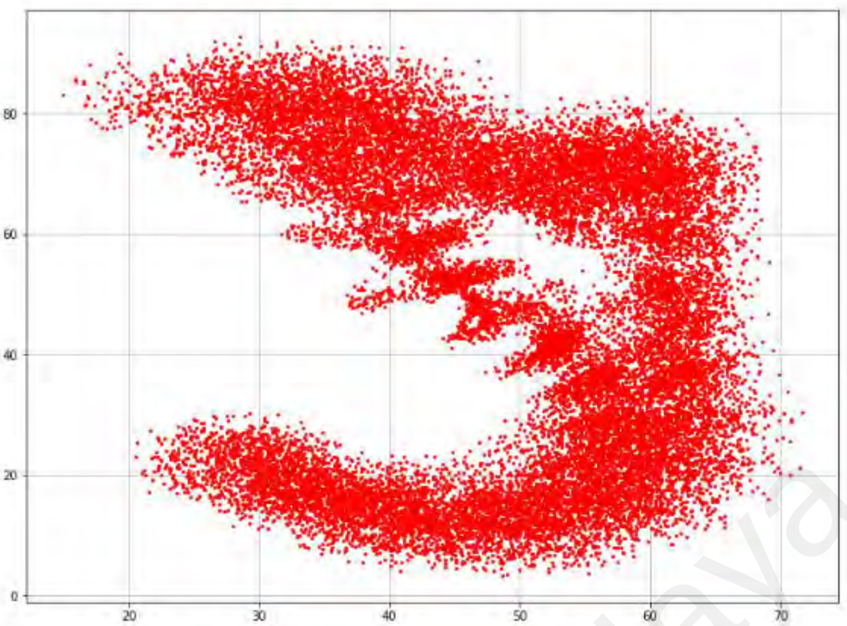


Figure 4.2: Distribution of the 34 constituent point primitives of every synthetic 2D illustrations in a 2D Euclidean space (Shape Category 1).

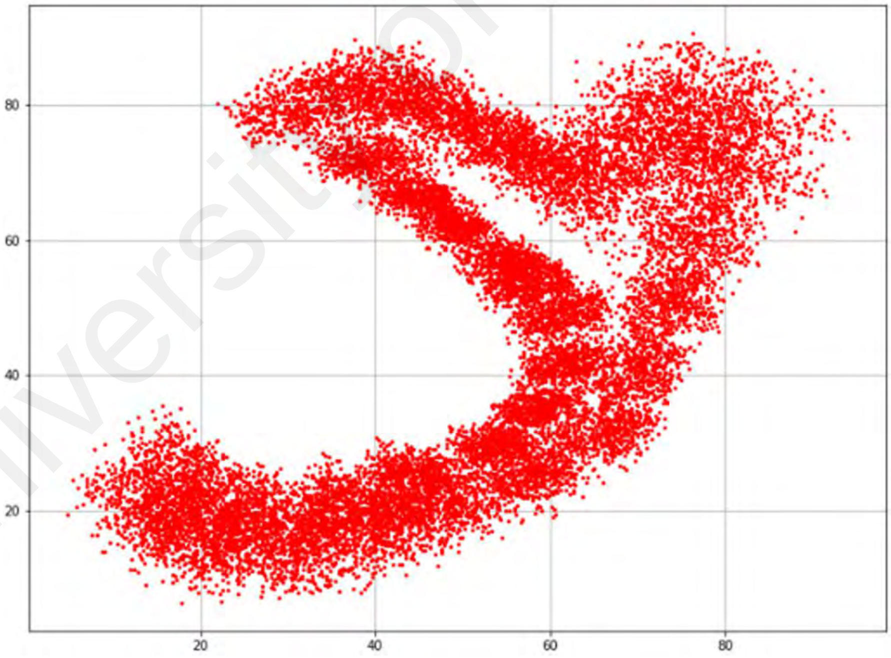


Figure 4.3: Distribution of the 34 constituent point primitives of every synthetic 2D illustrations in a 2D Euclidean space (Shape Category 2).

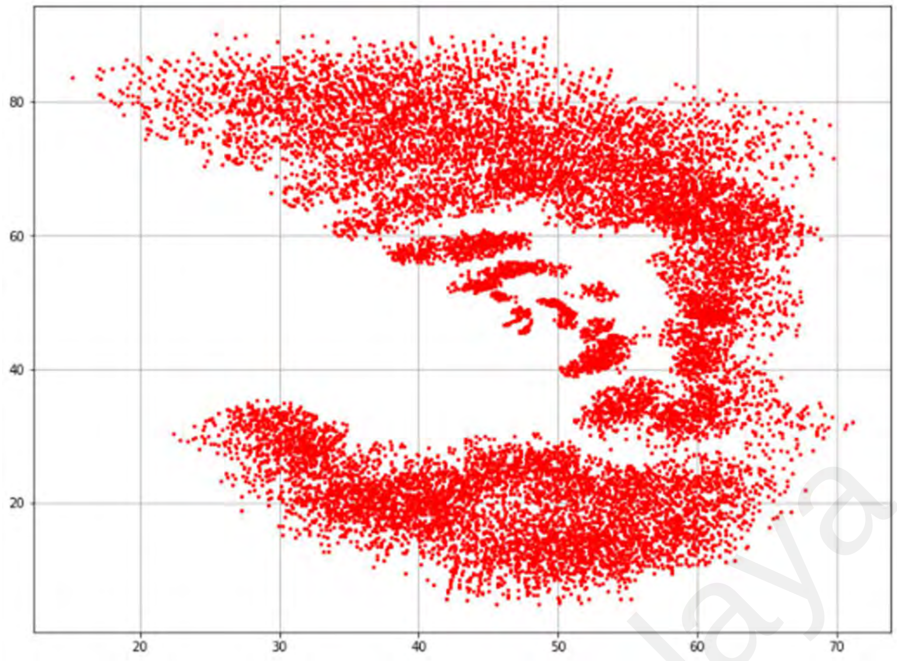


Figure 4.4: Distribution of the 34 constituent point primitives of every synthetic 2D illustrations in a 2D Euclidean space (Shape Category 3).

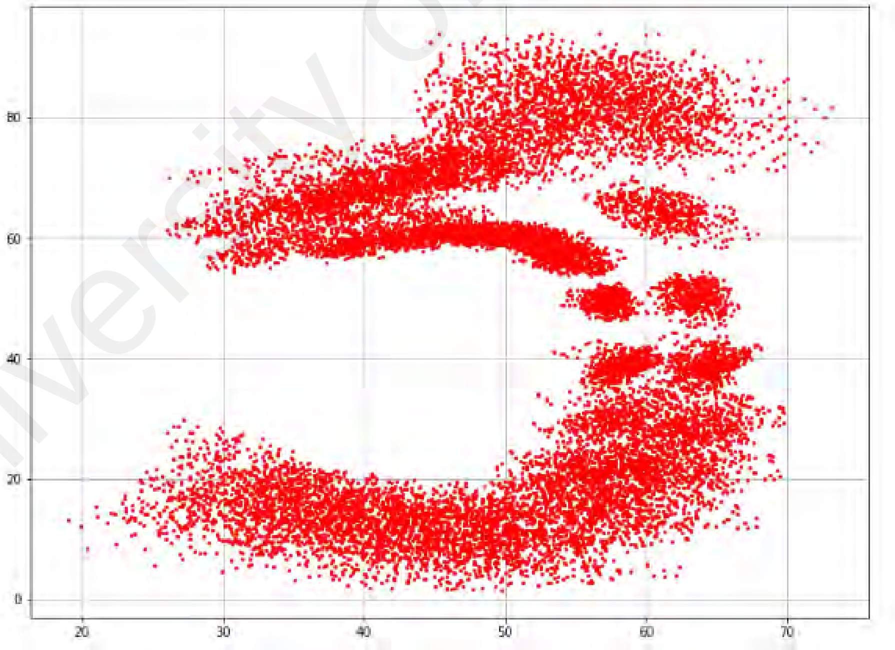


Figure 4.5: Distribution of the 34 constituent point primitives of every synthetic 2D illustrations in a 2D Euclidean space (Shape Category 4).

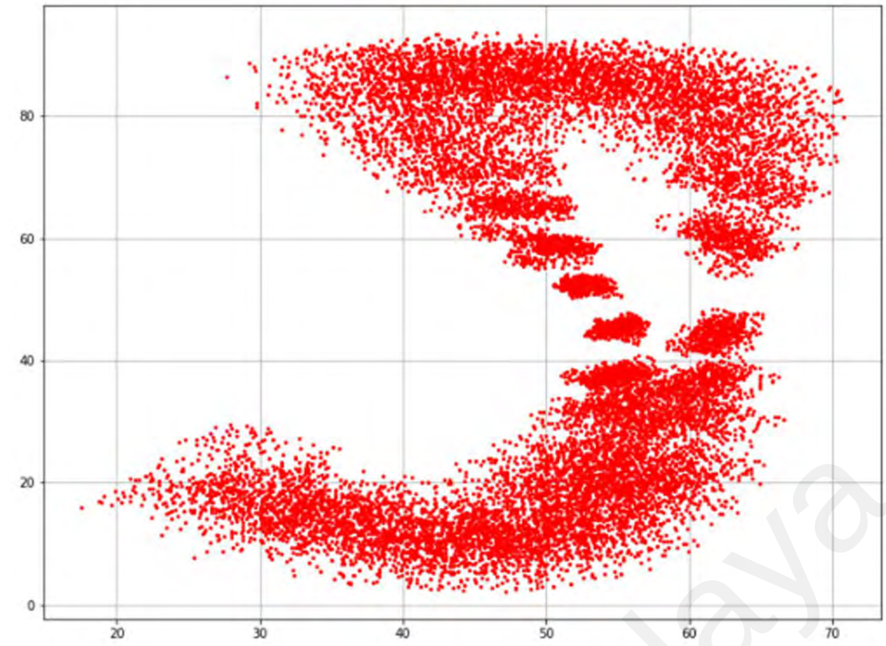


Figure 4.6: Distribution of the 34 constituent point primitives of every synthetic 2D illustrations in a 2D Euclidean space (Shape Category 5).

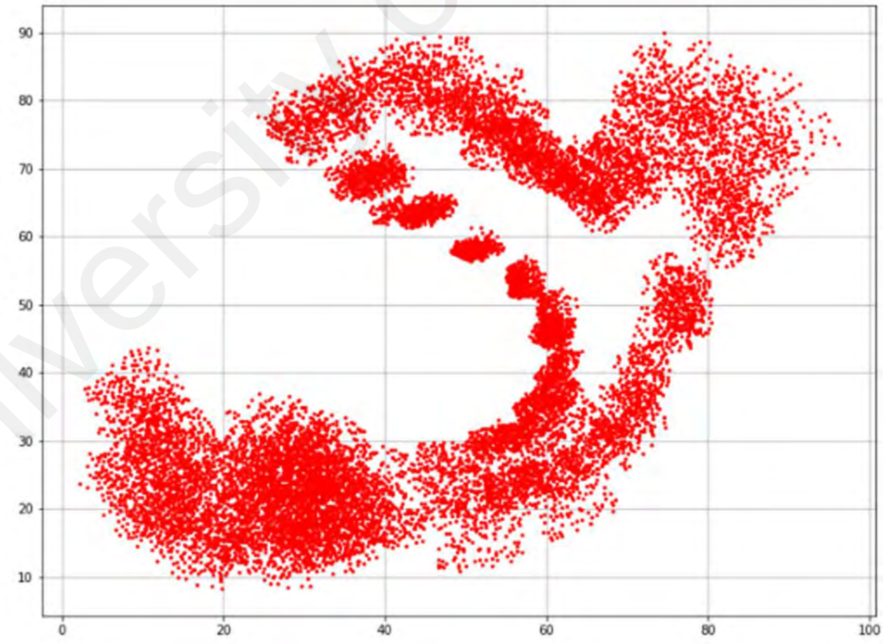


Figure 4.7: Distribution of the 34 constituent point primitives of every synthetic 2D illustrations in a 2D Euclidean space (Shape Category 6).

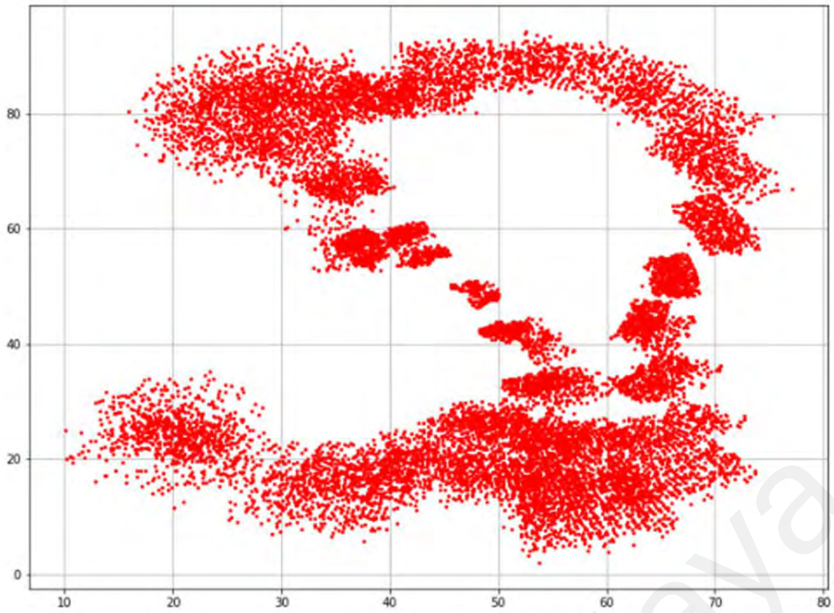


Figure 4.8: Distribution of the 34 constituent point primitives of every synthetic 2D illustrations in a 2D Euclidean space (Shape Category 7).

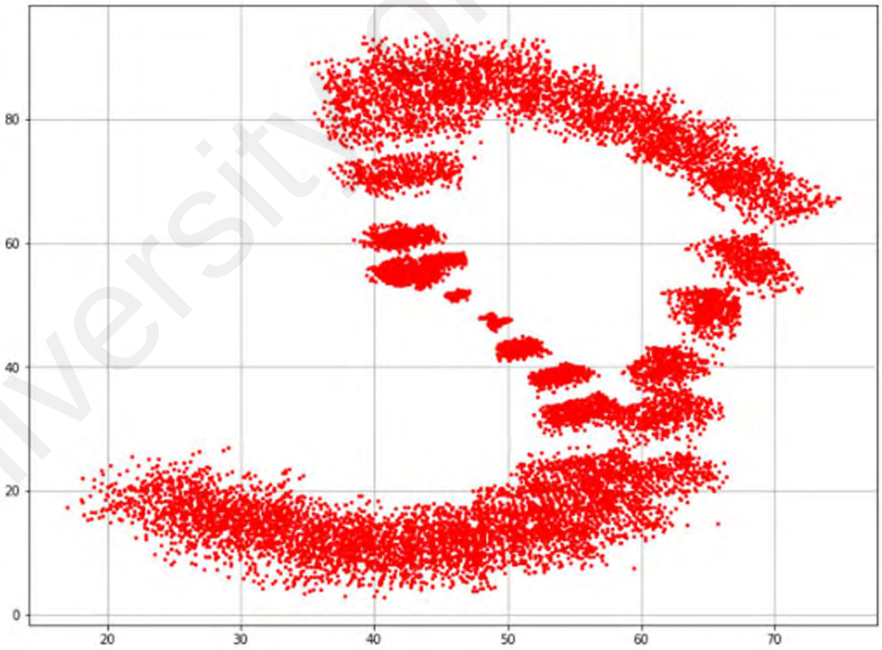
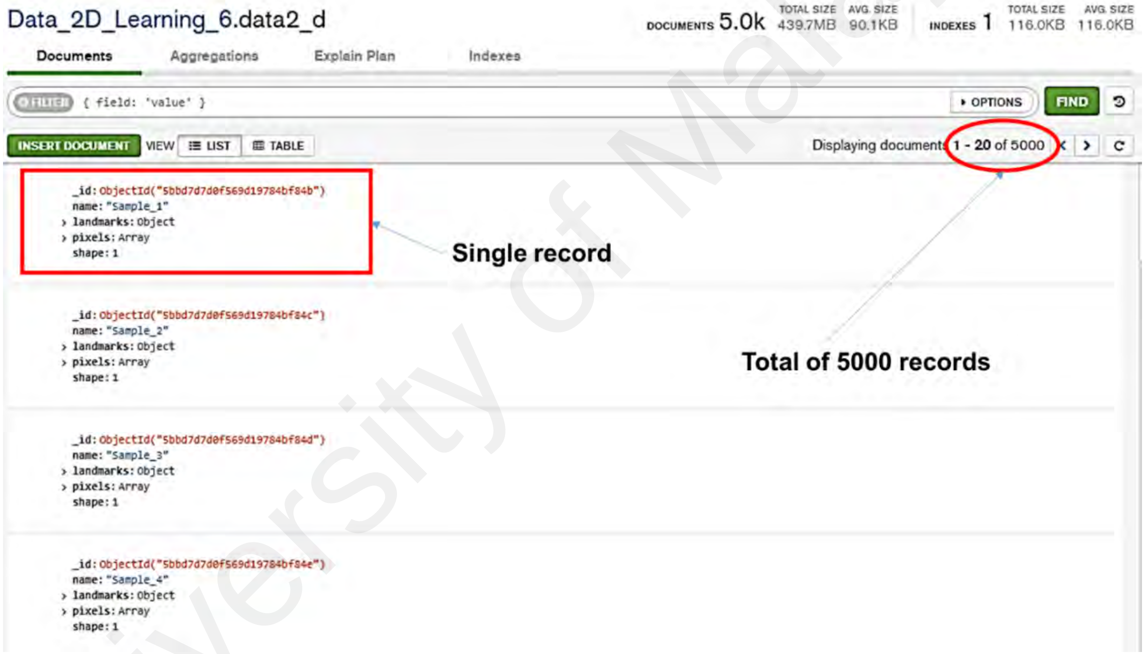


Figure 4.9: Distribution of the 34 constituent point primitives of every synthetic 2D illustrations in a 2D Euclidean space (Shape Category 8).

4.3.2 MongoDB database

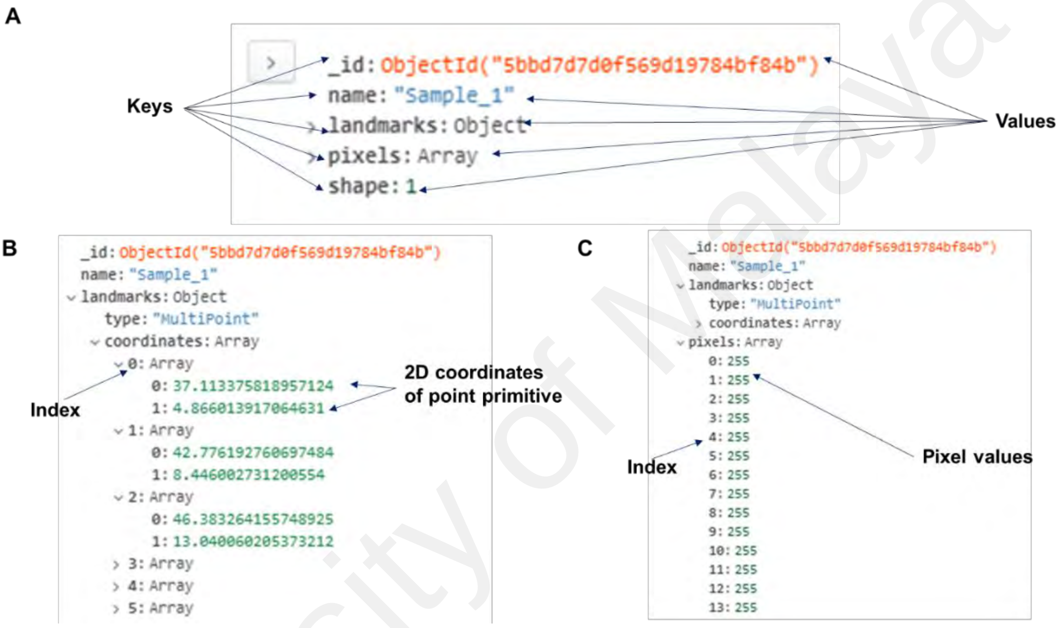
In this study, a NoSQL database, MongoDB was developed to store the details of all the 5000 synthetic illustration training sets such as the illustration name, pixels values of illustrations, 2D coordinates of 34 point primitives that constitute every illustration and shape category. The data of the training sets stored in the MongoDB can be visualized via an associated software, MongoDB Compass (Figure 4.10).



```
_id: ObjectId("5bbd7d7d0f569d19784bf84b")
  name: "Sample_1"
> landmarks: Object
> pixels: Array
  shape: 1

  _id: ObjectId("5bbd7d7d0f569d19784bf84c")
  name: "Sample_2"
> landmarks: Object
> pixels: Array
  shape: 1

  _id: ObjectId("5bbd7d7d0f569d19784bf84d")
  name: "Sample_3"
> landmarks: Object
> pixels: Array
  shape: 1

  _id: ObjectId("5bbd7d7d0f569d19784bf84e")
  name: "Sample_4"
> landmarks: Object
> pixels: Array
  shape: 1
```

Figure 4.10: MongoDB Compass that shows details of the stored training sets.

In the MongoDB Compass, the details of the training sets are presented as five key-value pairs of data for each record (Figure 4.11A). The "_id" key is auto generated by MongoDB whenever a new record is created. The "name" and the "shape" hold the information of the illustration name and the shape category as a simple string and a numeric number respectively. The 2D coordinates of the 34 point primitives are stored in "landmark" key and they are represented as coordinate arrays (Figure 4.11B). The term "landmark" is adopted as the key to store the 2D coordinates because all the 2D point primitives of each synthetic 2D illustrations are deployed as the landmark features to label the training sets in the supervised machine learning completed in this study (Refer to Chapter 3, Section 3.4). The pixel values of each illustration record is held by the "pixels" key and they are represented as an array of integer elements ranging from 0-255 (Figure 4.11C).



```
_id: ObjectId("5bbd7d7d0f569d19784bf84b")
name: "Sample_1"
landmarks: Object
pixels: Array
shape: 1
```

```
_id: ObjectId("5bbd7d7d0f569d19784bf84b")
name: "Sample_1"
landmarks: Object
   type: "MultiPoint"
   coordinates: Array
      0: Array
         0: 37.113375818957124
         1: 4.866013917064631
      1: Array
         0: 42.776192760697484
         1: 8.446002731200554
      2: Array
         0: 46.383264155748925
         1: 13.040060205373212
      3: Array
      4: Array
      5: Array
```

```
_id: ObjectId("5bbd7d7d0f569d19784bf84b")
name: "Sample_1"
landmarks: Object
   type: "MultiPoint"
   coordinates: Array
pixels: Array
   0: 255
   1: 255
   2: 255
   3: 255
   4: 255
   5: 255
   6: 255
   7: 255
   8: 255
   9: 255
   10: 255
   11: 255
   12: 255
   13: 255
```

Figure 4.11: Example of data stored in MongoDB.

4.4 Landmark point detection on the input illustrations using the machine learning model

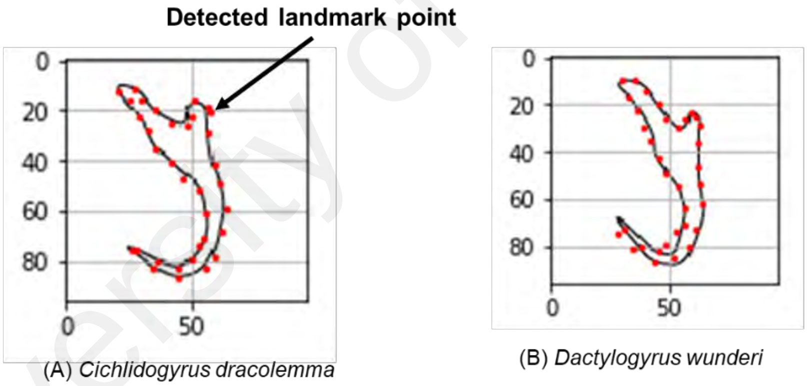
A machine learning model was trained by using the synthetic illustrations retrieved from the MongoDB database via the supervised learning process. The test accuracy of the supervised learning based on the different ratio of training set and validation set is presented in Table 4.2 below. Table 4.2 shows that the ANN model trained with a ratio of 70/30 produces highest train, validation, and test accuracy, which are 83.25%, 81.33%

and 80.1%, respectively. Hence, the ANN model trained with a ratio of 70/30 was adopted for the landmark detection of this study.

Table 4.2: Train, validation, and test accuracy of supervised learning based on different ratio of training set and validation set.

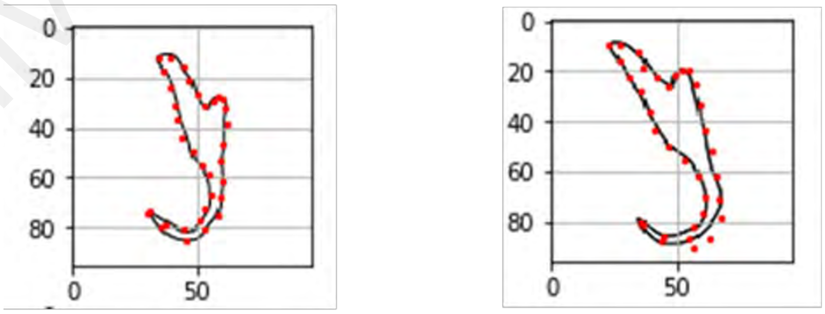
Ratio (Training set / validation set)	Train accuracy (%)	Validation accuracy (%)	Test accuracy (%)
80/20	80.78	79.25	77.5
70/30	83.25	81.33	80.1
60/40	80.00	76.37	75.10

Figure 4.12 show the landmark detection on the monogenean anchors of four selected species which share the similar shape feature of Category 1 whereas Figure 4.13 shows another eight selected monogenean species with different shape feature ranging from Category 1 to 8, respectively.



(A) *Cichlidogyrus dracolemma*

(B) *Dactylogyrus wunderi*



(C) *Cichlidogyrus raeymaekersi*

(D) *Cichlidogyrus aspiralis*

Figure 4.12: Machine detected landmarks on 2D illustrations of four selected monogenean anchors from Gusev (1973), Steenberge et al. (2015), Rahmouni et al. (2017) and Řehulková et al. (2013) that share similar shape feature.

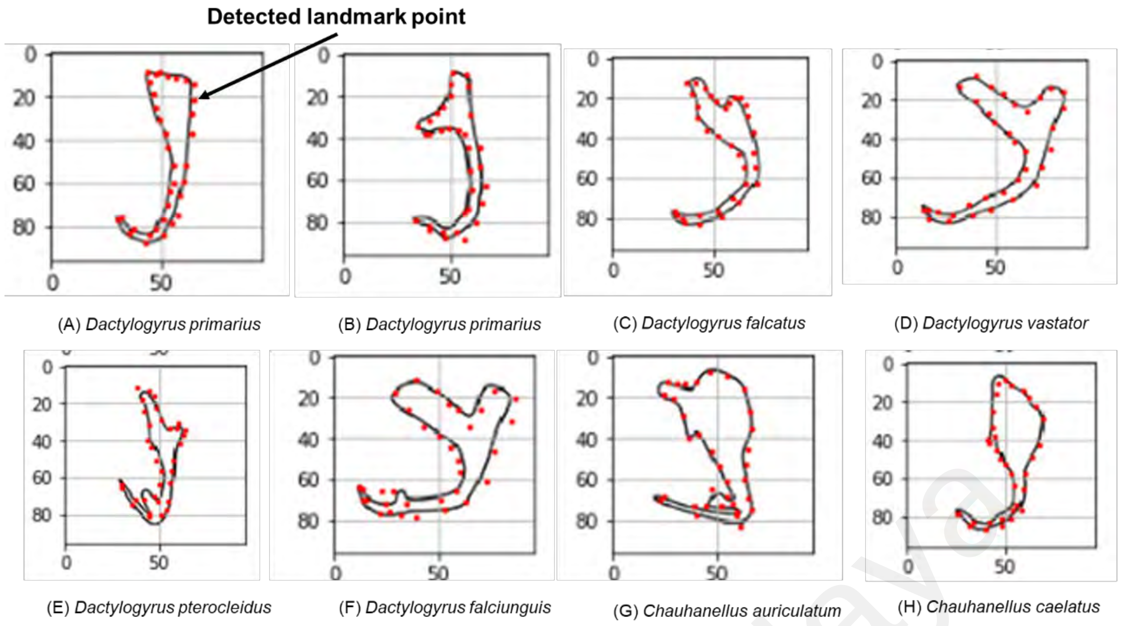


Figure 4.13: Machine detected landmarks on 2D illustrations of eight selected monogenean anchors from Gusev (1973) and Lim (1994).

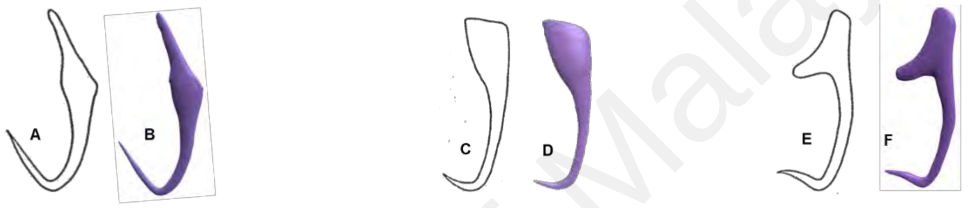
In general, the Figure 4.12 and Figure 4.13 show positive results of the landmark detection by having most of the landmark points (red dots) localized along the edge of the input 2D illustrations of the monogenean anchors. A more detailed evaluation of the efficiency of the machine learning model on landmark detection was done by calculating the Euclidean distance between the landmark points and each of their corresponding control points on the input illustrations. The evaluation result will be presented in Section 4.6.

4.5 3D models derived from deformation of the generic 3D model

In this section, the 3D models of monogenean anchors derived from the deformation of the generic 3D model through the manual approach and the automated 3D modelling pipeline are presented.

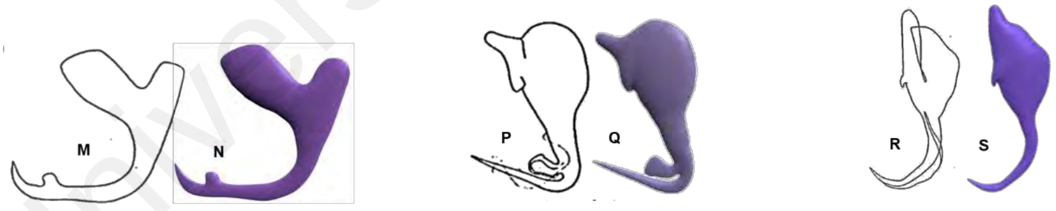
4.5.1 Deformed generic 3D model using the manual approach

Figure 4.14C-4.14S show the eight original 2D illustrations of anchors of varying complexities and their derived 3D forms constructed by deforming the generic 3D anchor model (Fig. 4.14B) through the manual deformation approach. The generic 3D anchor model could be deformed to form extrusions and morphologically different inner root as in Figure 4.14L & Figure 4.14N. This generic 3D anchor model can be deformed into complex anchor shapes because of the additional point primitives at the relevant sites on the 3D model.



(A-B) ***Bivaginogyrus obscurus*** **(Generic shape)** **(C-D)** ***Dactylogyrus primarius*** **(E-F)** ***Pellucidhaptor merus***

(G-H) ***Dactylogyrus falcatus*** **(I-J)** ***Dactylogyrus vastator*** **(K-L)** ***Dactylogyrus pterocleidus***



(M-N) ***Dactylogyrus falciunguis*** **(P-Q)** ***Chauhanellus auriculatum*** **(R-S)** ***Chauhanellus caelatus***

Figure 4.14: 2D anchor templates (unshaded) and corresponding 3D models (coloured).

The generic 3D model and the derived new 3D shapes have been rotated through 360° to reveal their 3-dimensional nature. Herein only the rotated views for the 3D anchor model of *Chauhanellus auriculatum* are shown (Figure. 4.15). Figure 4.15 also shows the different views of the 3D anchor in different orientations. This further elaborates the versatility of the newly developed generic 3D model. It should be noted that the generic 3D model (Figure. 4.14B) and the derived 3D models were smoothened using the Catmull-Clark smoothen modifier in Blender to reduce the angularities caused by the rectangular building blocks. The jagged edges of the generic and derived 3D models caused by the inherent rendering problem in Blender are reduced using the antialiasing option in Blender but not completely eliminated.

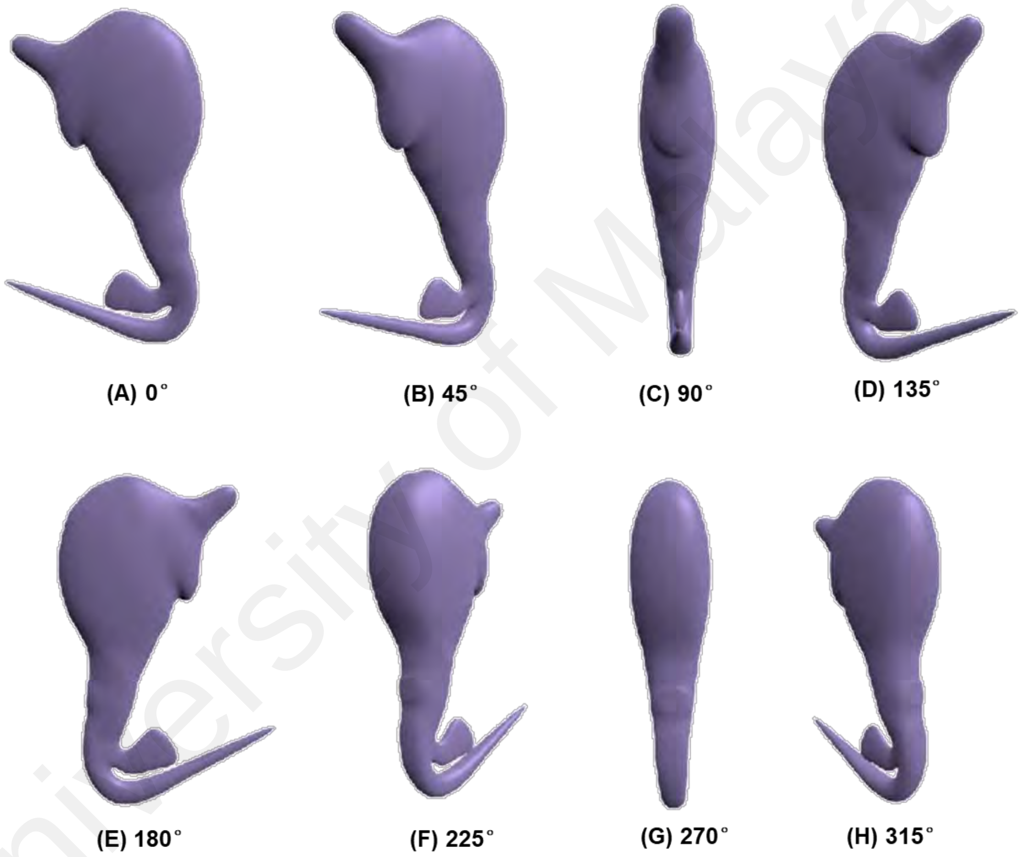
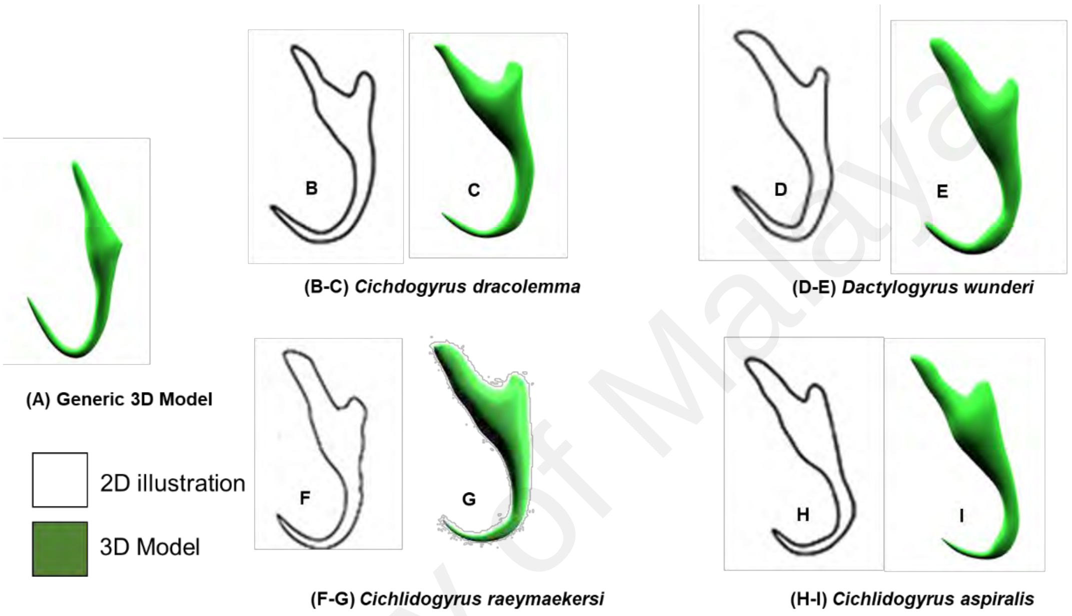


Figure 4.15: Views of the 3D anchor of *Chauhanellus auricalatum* in different degrees of rotation (anti-clockwise) in the z-axis.

4.5.2 Deformed generic 3D model using the automated 3D modelling pipeline

Figure 4.24 shows a group of 3D models (Figure 4.16 B- E) produced by the automated 3D modelling pipeline. These resulting 3D models share the common shape features (Category 1) and the reason this category of shapes was used as the test objects is because they are relatively more commonly found in the surveyed publications. Apparently, the predicted landmark points obtained from the machine learning model could be used to drive the deformation of the generic 3D model (Figure 4.16A) to derive the plausible 3D shapes (Figure 4.16C, 4.16E, 4.16G & 4.16I) that can mimic the shape presented in their corresponding original 2D illustrations respectively (Figure 4.16B, 4.16D, 4.16F & 4.16H).



(B-C) *Cichdogyrus dracolemma*

(D-E) *Dactylogyrus wunderi*

(F-G) *Cichlidogyrus raeymaekersi*

(H-I) *Cichlidogyrus aspiralis*

Figure 4.16: The generic 3D model and the input illustrations of anchor (unshaded) along with their corresponding 3D models (coloured).

Figure 4.17 shows another group of 3D anchors generated by also using the machine learning empowered automated 3D modelling pipeline developed in this study. Instead of sharing the common shape feature, the 3D models vary from one another by having their distinctive shape characteristic which is very different from one another. By comparing each of the resulting 3D models with their corresponding original input illustrations, the automated 3D modelling pipeline has shown its capability to deform the generic 3D model (Figure 4.17A) into multiple shape categories of anchor. In general, the overall body shape of the resulting 3D models resembles the 2D shape as presented in their corresponding 2D illustrations. Nevertheless, the automated 3D modelling pipeline does not work well on modelling the extrusion parts of few anchors (Figure 4.17K, & Figure 4.17O). A closer inspection on the shape discrepancy presented by those anchors is shown in Figure 4.18. The automated 3D modelling pipeline shows its limitation to derive the fan-like extrusion structure found in the anchors of *Dactylogyrus peterocleidus* and *Chauhanellus auriculatum*.

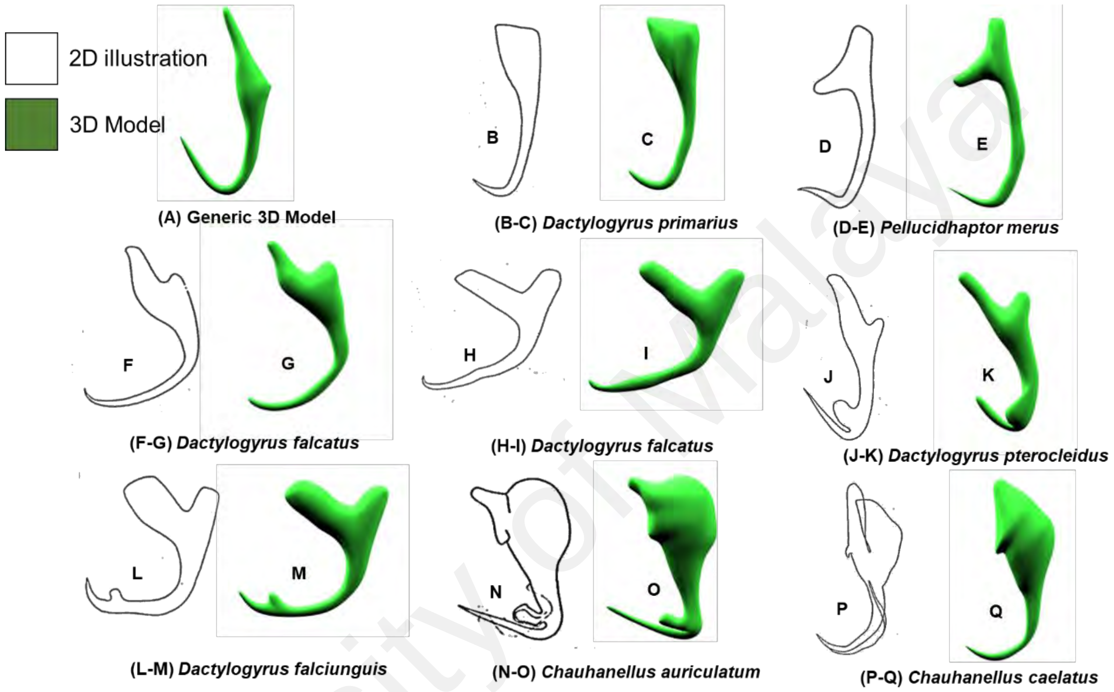


Figure 4.17: Generic 3D model and the input illustrations of anchor along with their corresponding 3D models (coloured).

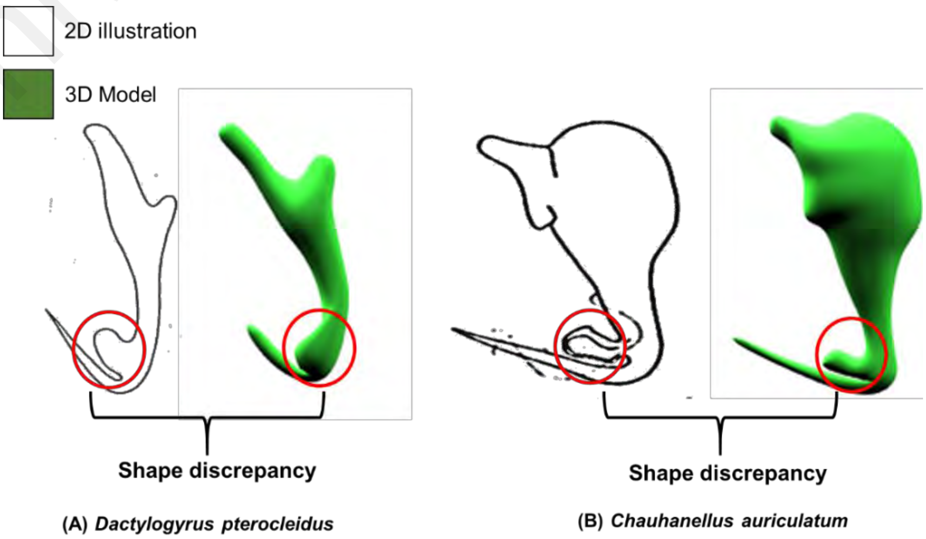


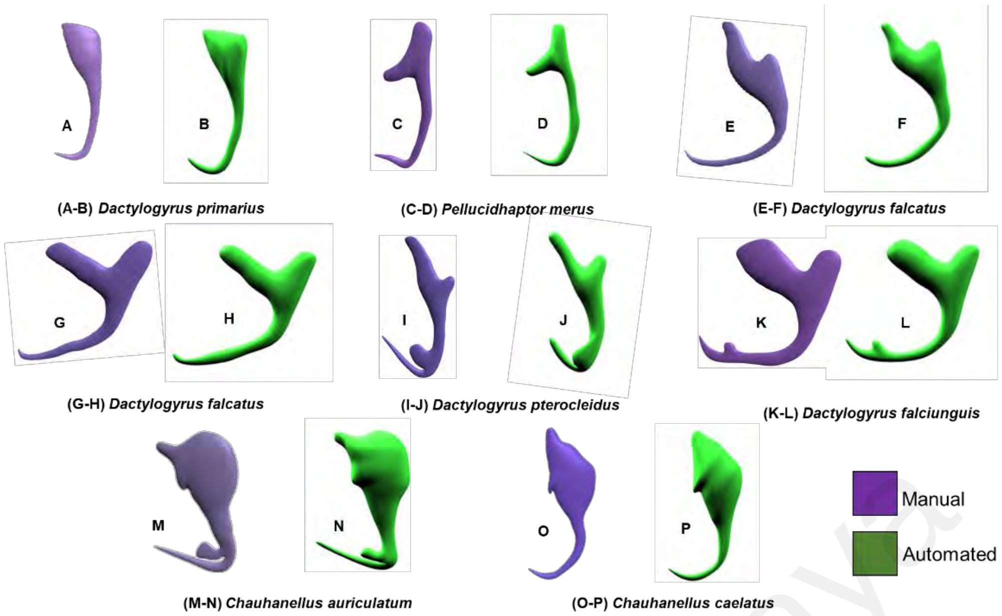
Figure 4.18: Discrepancy of 3D shape found in the extrusion part.

4.6 Evaluation of the automated 3D modelling pipeline

In this section, the results of evaluation on the efficiency of the automated 3D modelling pipeline is presented by showing the details of comparison of the 3D anchors produced from the manual deformation and the automated 3D modelling pipeline and quantitative evaluation on the automated 3D modelling pipeline.

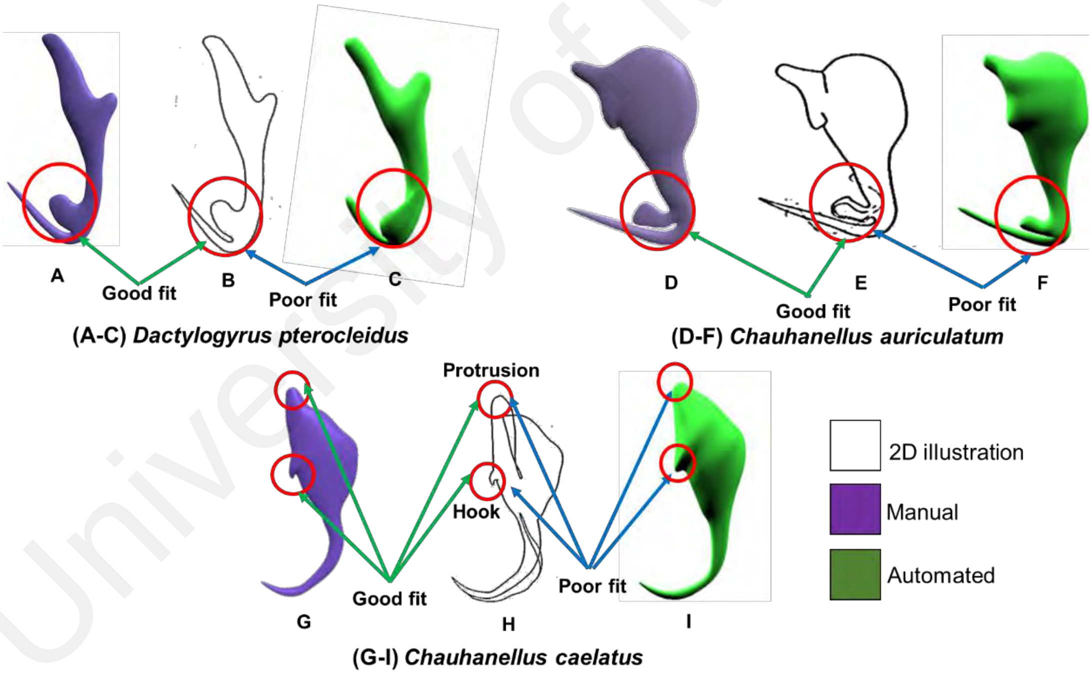
4.6.1 Comparison of the deformed 3D anchors

A comparison was done on the 3D anchors derived from the manual deformation approach and the automated 3D modeling pipeline (Figure 4.19). Overall, the resulting 3D models produced from both manual and automated approach possess the entire body shape that resemble the 2D shapes in the original 2D illustrations. However, the fan-like extrusion structure found in the 3D anchors of *Dactylogyrus. pterocleidus* and *Chauhanellus auriculatum* derived from the manual deformation approach are much more natural and accurate (Figure 4.20A & D). In addition, the hook and a protrusion on top of the 3D anchor of *Chauhanellus caelatus* derived by the manual deformation approach looked more realistic than the one created by the automated 3D modelling pipeline (Figure 4.20G).



(A-B) *Dactylogyrus primarius*
(C-D) *Pellucidhaptor merus*
(E-F) *Dactylogyrus falcatus*
(G-H) *Dactylogyrus falcatus*
(I-J) *Dactylogyrus pterocleidus*
(K-L) *Dactylogyrus falciunguis*
(M-N) *Chauhanellus auriculatum*
(O-P) *Chauhanellus caelatus*

Figure 4.19: Comparison of the derived 3D models from manual deformation approach (purple colored) and the automated 3D modelling pipeline (green colored).



(A-C) *Dactylogyrus pterocleidus*
(D-F) *Chauhanellus auriculatum*
(G-I) *Chauhanellus caelatus*

Figure 4.20: A closer look on the comparison of three selected 3D anchors derived from manual deformation approach (purple colored) and the automated 3D modelling pipeline (green colored) along with their corresponding 2D illustrations (without shaded).

4.6.2 Quantitative evaluation on the automated 3D modelling pipeline

In Section 4.6.1, a rudimentary comparison between the 3D anchors derived from the manual deformation approach and the automated 3D modelling pipeline was presented by showing how well each of the resulting 3D models fit the shape presented in each of their corresponding 2D illustrations. Such qualitative comparison depends on a subjective visual inspection to determine the quality of the resulting 3D models. In this section, the results of quantitative evaluation on the efficiency of the automated 3D modelling pipeline in modelling all the anchors of 12 selected monogenean species are presented. This quantitative evaluation is based on a comparative approach to examine the discrepancy between a set of 34 control points and the 34 predicted landmark points on the 2D illustrations of anchor. The results of quantitative evaluation were presented by showing calculated Euclidean distances between the 34 predicted landmark points and the 34 control points assigned along the edges of the input illustrations of anchor.

The Euclidean distance calculation was done for all the 12 selected monogenean species and the results are presented in Figures 4.21- 4.32. The presentation of the results is divided into four sections. The first section (Figure. 4.21A- 4.32A) shows the tabulated Euclidean distances with each of them indexed by a point index starting from 0 to 33. In the second section (Figure 4.21B – 4.32B), the line plots that show the distribution of the Euclidean distances for each species across the point indices were presented and at the same time several critical regions were highlighted in each of the plots. The third (Figure 4.21C – 4.32C) and fourth section (Figure 4.21D- 4.32D) show the sites of high morphological variations on the 2D input illustrations and their corresponding derived 3D model respectively. These are the regions where further examination was done to check for the existence of the shape distortion in a derived 3D model. The observations on those sites of high morphological variations and how they affect the resulting 3D model of all the 12 species are summarized in Table 4.3 below:

Table 4.3: Summary of the shape distortion on the sites of high morphological variations.

Species	Observation on sites of high morphological variations	
Cichlidogyrus dracolemma (Figure 4.21)	Site I	• Predicted points diverted downwards from control points. • Effect – The inner root of the 3D anchor is slightly bended down.
	Site II	• Predicted point diverted downwards from control points. • Predicted points still positioned on the edge • Effect – No noticeable shape distortion.
	Site III	• Predicted points diverted downwards from control points. • Effect – tip is thinner than expected.
	Site IV	• Point index-22 is diverted upwards from the control points. • The body shape of this region become thicker than expected.
	Site V	• Predicted point diverted downwards from control points. • The body shape of this region is slightly tilted down.
	Site VI	• Predicted point diverted downwards from control points. • The body shape of this region is slightly tilted down.
	Site VII	• Predicted point diverted downwards from control points. • Predicted points still positioned on the edge • Effect – No noticeable shape distortion.
Dactylogyrus wunderi (Figure 4.22)	Site I	• Predicted points and control points are almost aligned with one another. • Effect – No noticeable shape distortion.

Table 4.3, continued.

Species	Observation on sites of high morphological variations	
Dactylogyrus wunderi (Figure 4.22)	Site II	• Predicted points is slightly diverted downward from the control points. • Predicted points still positioned on the edge • Effect – No noticeable shape distortion.
	Site III	• Predicted point diverted away from control point in downward direction. • Effect – the anchor tip tilted down.
	Site IV	• Point index-22 is diverted upwards from the control points. • The body shape of this region become thicker than expected.
	Site V	• Predicted point diverted downwards from control points. • The body shape of this region is slightly tilted down.
	Site VI	• Predicted point diverted downwards from control points. • The body shape of this region is slightly tilted down.
	Site VII	• Predicted points is slightly diverted away from the control points. • Predicted points still positioned on the edge • Effect – No noticeable shape distortion.
Cichlidogyrus raeymaekersi (Figure 4.23)	Site I	• Predicted points diverted from the control points in right direction. • Effect – Outer root of 3D anchor is slight shifted to the right direction.
	Site II	• Predicted points diverted downwards from the control points. • Predicted points still positioned along the edge. • Effect – no noticeable shape distortion.
	Site III	• Predicted points diverted far from the control points in upward direction. • Effect – anchor tip tilted upwards.
	Site IV	• Predicted points diverted inwards from the control points. • Effect – Body shape of that region become thinner than expected.
	Site V	• Predicted points diverted from the control points in upward direction. • Predicted points still positioned on the edge • Effect – No noticeable shape distortion.
	Site VI	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.

Table 4.3, continued.

Species	Observation on sites of high morphological variations	
Cichlidogyrus raeymaekersi (Figure 4.23)	Site VII	• Predicted points diverted from the control points in right direction. • Effect – Inner root tilted to the right.
Cichlidogyrus aspiralis (Figure 4.24)	Site I	• Low discrepancy between predicted points and control points. Effect – No noticeable shape distortion.
	Site II	• Predicted points diverted downwards from the control points. • Predicted points still positioned along the edge • Effect – no noticeable shape distortion.
	Site III	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site IV	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site V	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site VI	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site VII	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
Dactylogyrus primarius (Figure 4.25)	Site I	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site II	• Predicted points and control points are almost aligned with one another. • Effect – No noticeable shape distortion.
	Site III	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site IV	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site V	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site VI	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site VII	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.

Table 4.3, continued.

Species	Observation on sites of high morphological variations	
Pellucidhaptor merus (Figure 4.26)	Site I	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site II	• Predicted points diverted from the control points in upward direction. • Effect – The body shape of that region is slightly tilted upwards.
	Site III	• Predicted points significantly tilted downward from the control point. • The anchor tip become thinner than expected.
	Site IV	• Predicted points diverted from the control point in right direction. • The body shape of that region is tilted to the right.
	Site V	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site VI	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site VII	• Predicted points diverted from the control points in downward direction. • Inner root is tilted down.
Dactylogyrus falcatus (Figure 4.27)	Site I	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site II	• Predicted points diverted from the control points in upward direction. • Predicted points still positioned along the edge. • Effect – no noticeable shape distortion.
	Site III	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site IV	• Predicted points diverted from the control points in downward direction. • Predicted points still positioned along the edge. • Effect – no noticeable shape distortion.
	Site V	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site VI	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.

Table 4.3, continued.

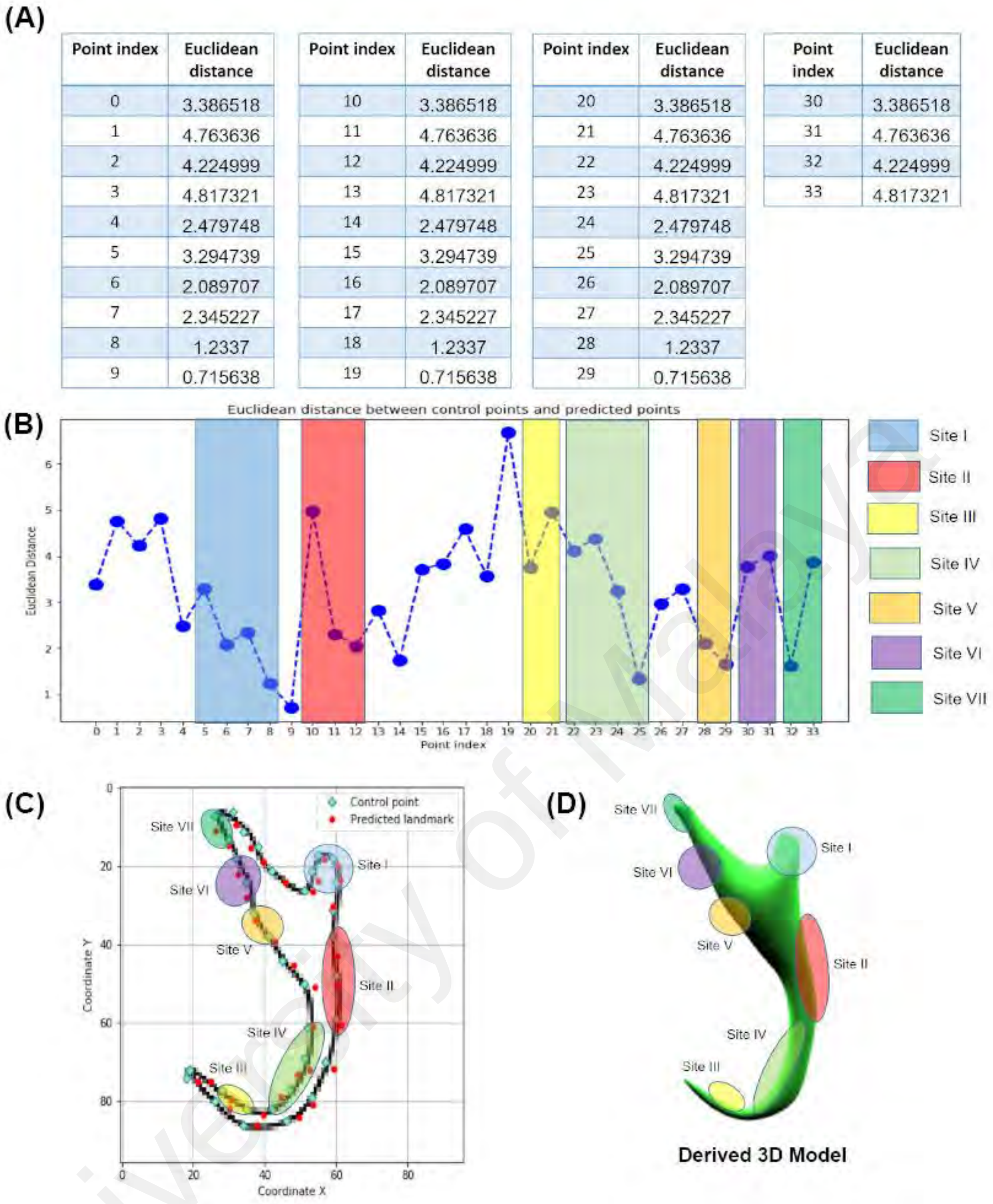
Species	**Observation on sites of high morphological variations**	
Dactylogyrus falcatus (Figure 4.27)	Site VII	- Low discrepancy between predicted points and control points. - Effect – No noticeable shape distortion.
Dactylogyrus vastator (Figure 4.28)	Site I	- Low discrepancy between predicted points and control points. - Effect – No noticeable shape distortion.
	Site II	- Predicted point diverted downwards from the control point. - Predicted point still positioned on the edge - Effect – No noticeable shape distortion.
	Site III	- Low discrepancy between predicted points and control points. - Effect – No noticeable shape distortion.
	Site IV	- Low discrepancy between predicted points and control points. - Effect – No noticeable shape distortion.
	Site V	- Predicted point diverted from the control point in left direction. - Effect – The body shape of that region is slightly tilted to the left.
	Site VI	- Predicted point diverted downwards from the control point. - Predicted point still positioned on the edge - Effect – No noticeable shape distortion.
	Site VII	- Predicted point diverted downwards from the control point. - Predicted point still positioned on the edge - Effect – No noticeable shape distortion.
Dactylogyrus pterocleidus (Figure 4.29)	Site I	- Low discrepancy between predicted points and control points. - Effect – No noticeable shape distortion.
	Site II	- Predicted point diverted downwards from the control point. - Predicted point still positioned on the edge - Effect – No noticeable shape distortion.
	Site III	- Predicted point diverted upward from the control point. - Effect – The anchor tip is tilted upward.
	Site IV	- Predicted points diverted leftwards from the control points. - Effect – Extrusion is seriously distorted.
	Site V	- Predicted point diverted upward from the control point. - Predicted point still positioned on the edge - Effect – No noticeable shape distortion.

Table 4.3, continued.

Species	Observation on sites of high morphological variations	
Dactylogyrus pterocleidus (Figure 4.29)	Site VI	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site VII	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
Dactylogyrus falciunguis (Figure 4.30)	Site I	• Predicted points diverted rightwards form the control points. • Effect – The outer root tilted to the right and the shape is thinner than expected.
	Site II	• Predicted point diverted downwards from the control point. • Predicted point still positioned on the edge • Effect – No noticeable shape distortion.
	Site III	• Predicted points diverted leftwards form the control points. • Effect – The extrusion tilted to the left.
	Site IV	• One predicted point in this region is diverted towards the right and another downwards from the control point. • Effect – No noticeable shape distortion.
	Site V	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site VI	• Predicted point diverted downwards from the control point. • Predicted point still positioned on the edge. • Effect – No noticeable shape distortion.
	Site VII	• Predicted point diverted downwards from the control point. • Predicted point still positioned on the edge. • Effect – No noticeable shape distortion.
Chauhanellus auriculatum (Figure 4.31)	Site I	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site II	• Predicted points diverted downwards from the control points. • Predicted points still positioned on the edge. • Effect - No noticeable shape distortion.
	Site III	• Predicted points diverted inwards from the control points. • Effect – The tip of the anchor is slightly shorter than expected.

Table 4.3, continued.

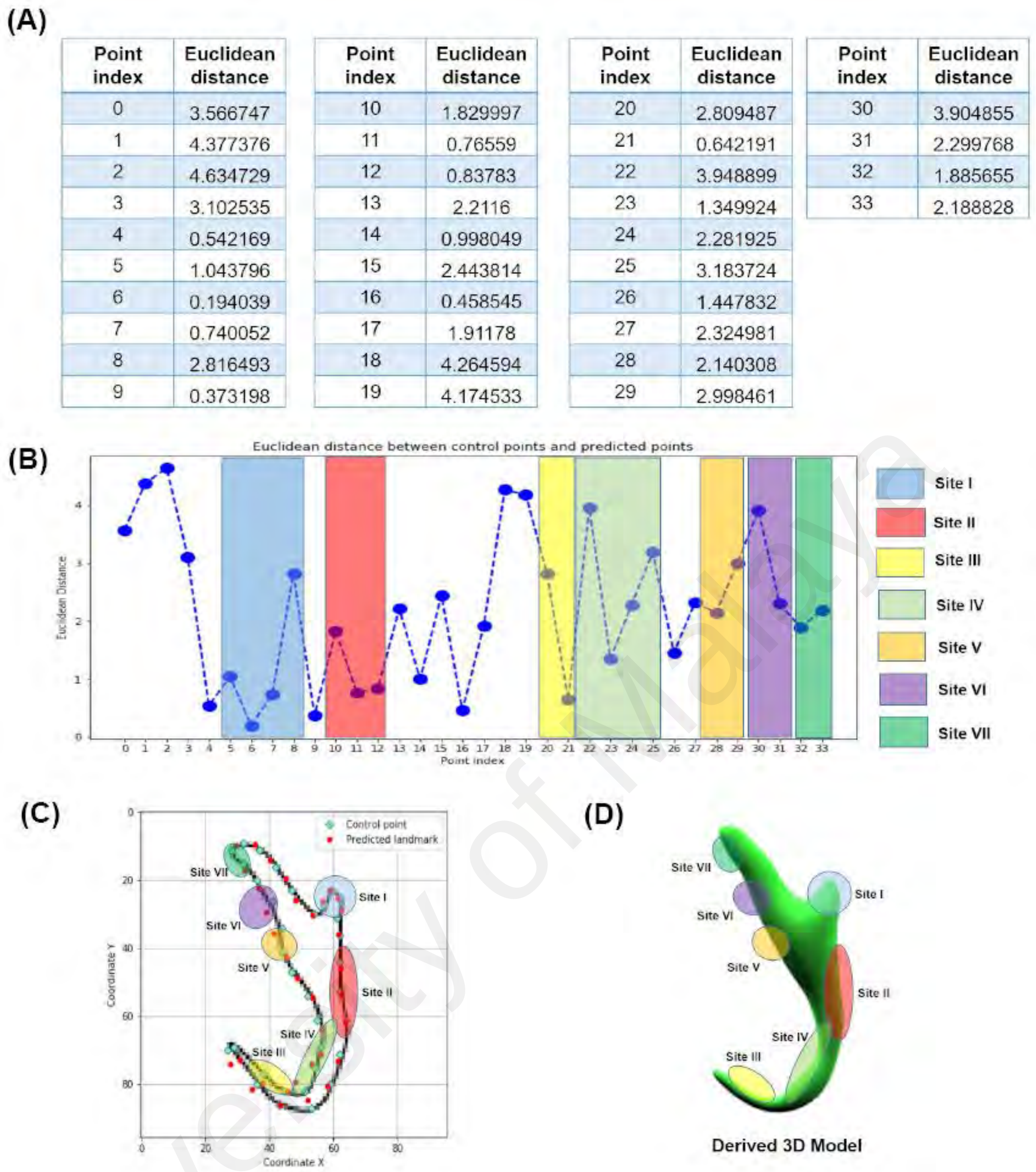
Species	**Observation on sites of high morphological variations**	
Chauhanellus auriculatum (Figure 4.31)	Site IV	• Two predicted points diverted downwards and another two diverted upwards from the control points. • Effect – Shape of extrusion is distorted.
	Site V	• Predicted point diverted upwards from the control point. • Predicted point still positioned on the edge • Effect – No noticeable shape distortion.
	Site VI	• Predicted points diverted downwards from the control points. • Effect – the extrusion of that region is slightly tilted down.
	Site VII	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
Chauhanellus caelatus (Figure 4.32)	Site I	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site II	• Predicted points diverted downwards from the control points. • Effect – Anchor tip is slightly bended down.
	Site III	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site IV	• Predicted points diverted towards the right. • Predicted points still positioned on the edge. • Effect – No noticeable shape distortion.
	Site V	• Low discrepancy between predicted points and control points. • Effect – No noticeable shape distortion.
	Site VI	• Predicted points diverted to towards the right. • Effect – Shape of the hook looked unnatural.
	Site VII	• Predicted point diverted downwards from the control point. • Effect – protrusion of anchor is bended down.



(A)

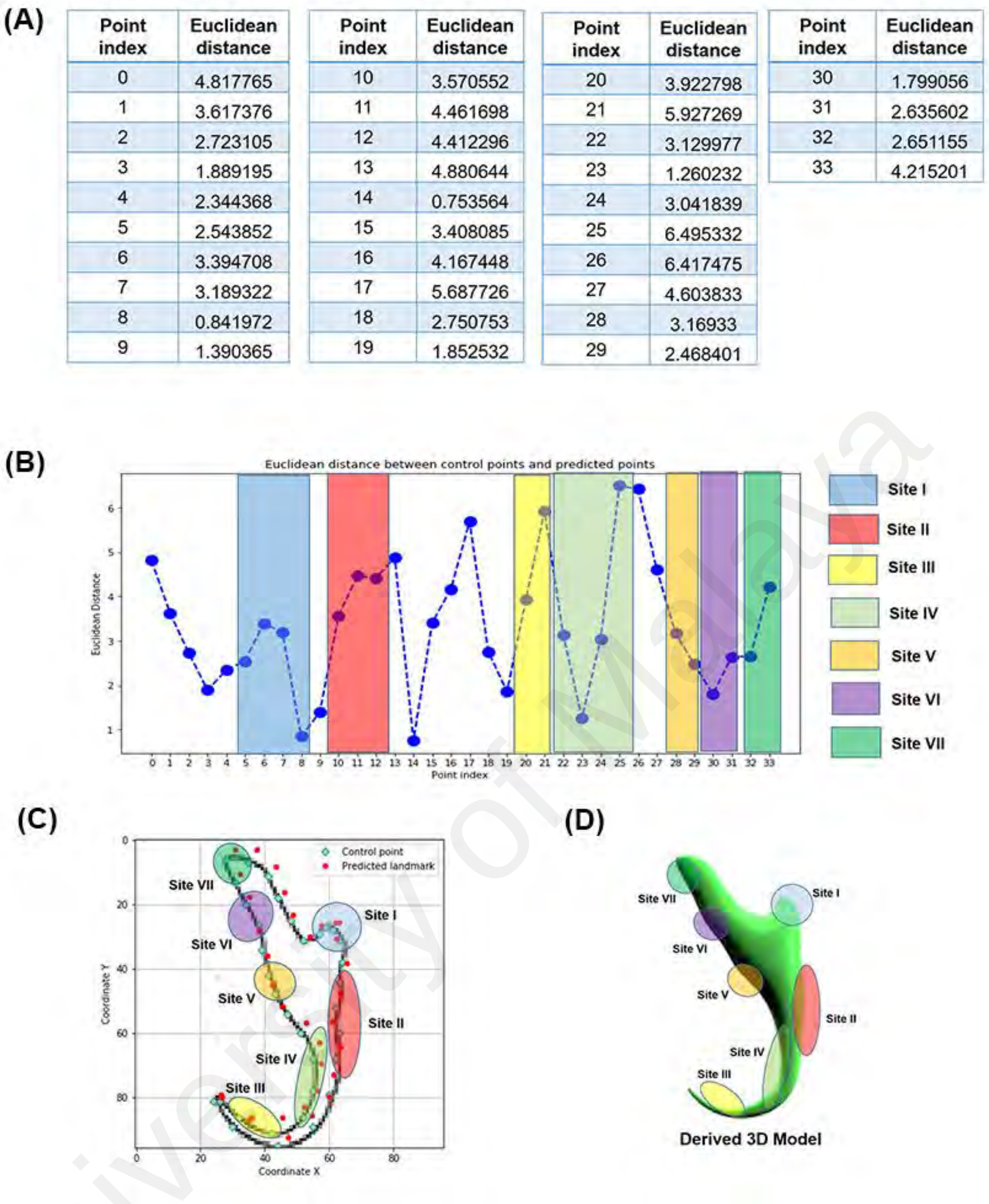
Point index	Euclidean distance	Point index	Euclidean distance	Point index	Euclidean distance	Point index	Euclidean distance
0	3.386518	10	3.386518	20	3.386518	30	3.386518
1	4.763636	11	4.763636	21	4.763636	31	4.763636
2	4.224999	12	4.224999	22	4.224999	32	4.224999
3	4.817321	13	4.817321	23	4.817321	33	4.817321
4	2.479748	14	2.479748	24	2.479748		
5	3.294739	15	3.294739	25	3.294739		
6	2.089707	16	2.089707	26	2.089707		
7	2.345227	17	2.345227	27	2.345227		
8	1.2337	18	1.2337	28	1.2337		
9	0.715638	19	0.715638	29	0.715638		

Figure 4.21: Quantitative analysis of shape accuracy of the derived 3D anchor (*Cichlidogyrus dracolemma*) based on Euclidean distance.



Point index	Euclidean distance
0	3.566747
1	4.377376
2	4.634729
3	3.102535
4	0.542169
5	1.043796
6	0.194039
7	0.740052
8	2.816493
9	0.373198
10	1.829997
11	0.76559
12	0.83783
13	2.2116
14	0.998049
15	2.443814
16	0.458545
17	1.91178
18	4.264594
19	4.174533
20	2.809487
21	0.642191
22	3.948899
23	1.349924
24	2.281925
25	3.183724
26	1.447832
27	2.324981
28	2.140308
29	2.998461
30	3.904855
31	2.299768
32	1.885655
33	2.188828

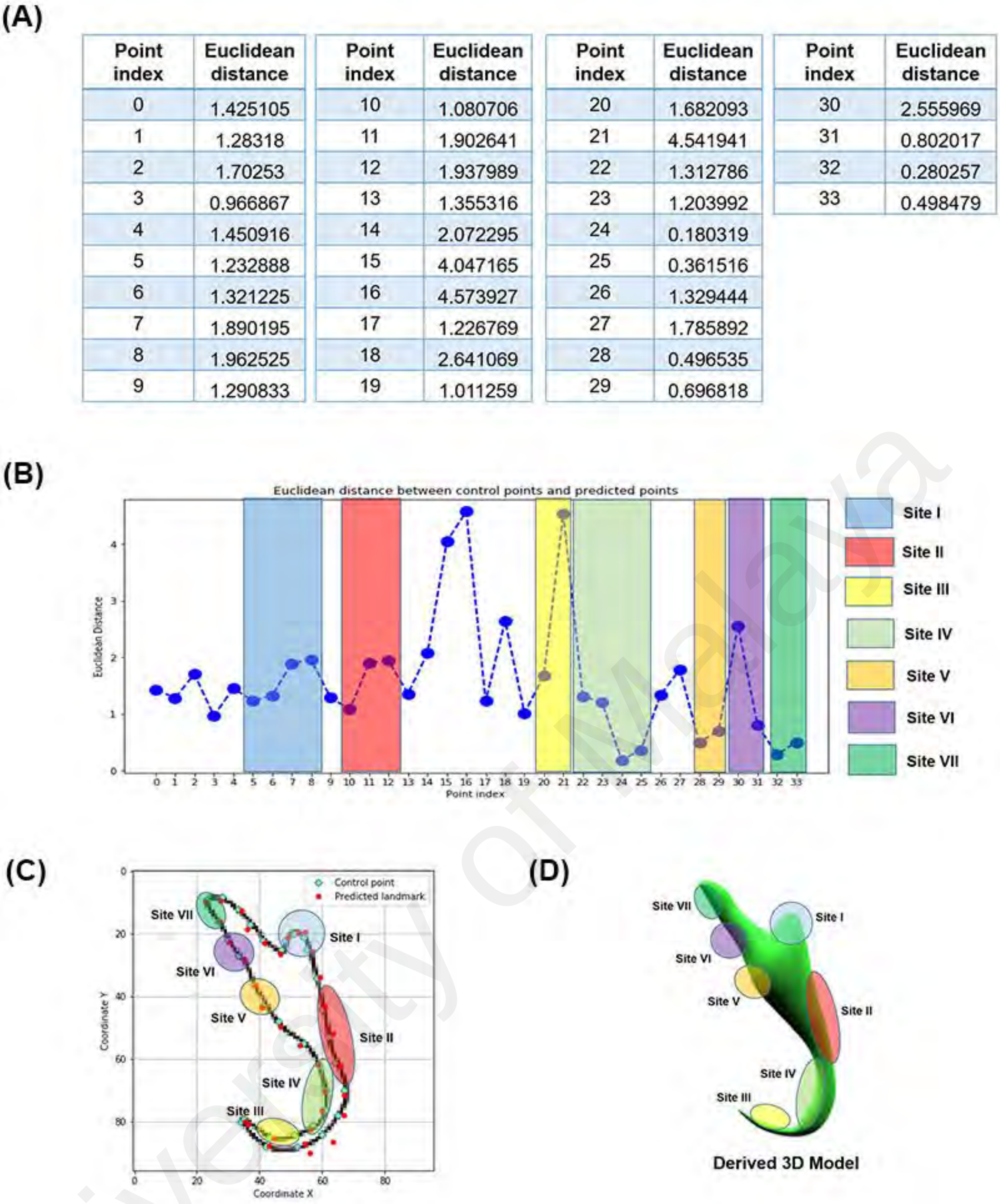
Figure 4.22: Quantitative analysis of shape accuracy of the derived 3D anchor (*Dactylogyrus wunderi*) based on Euclidean distance.


(A)

Point index	Euclidean distance
0	4.817765
1	3.617376
2	2.723105
3	1.889195
4	2.344368
5	2.543852
6	3.394708
7	3.189322
8	0.841972
9	1.390365
10	3.570552
11	4.461698
12	4.412296
13	4.880644
14	0.753564
15	3.408085
16	4.167448
17	5.687726
18	2.750753
19	1.852532
20	3.922798
21	5.927269
22	3.129977
23	1.260232
24	3.041839
25	6.495332
26	6.417475
27	4.603833
28	3.16933
29	2.468401
30	1.799056
31	2.635602
32	2.651155
33	4.215201

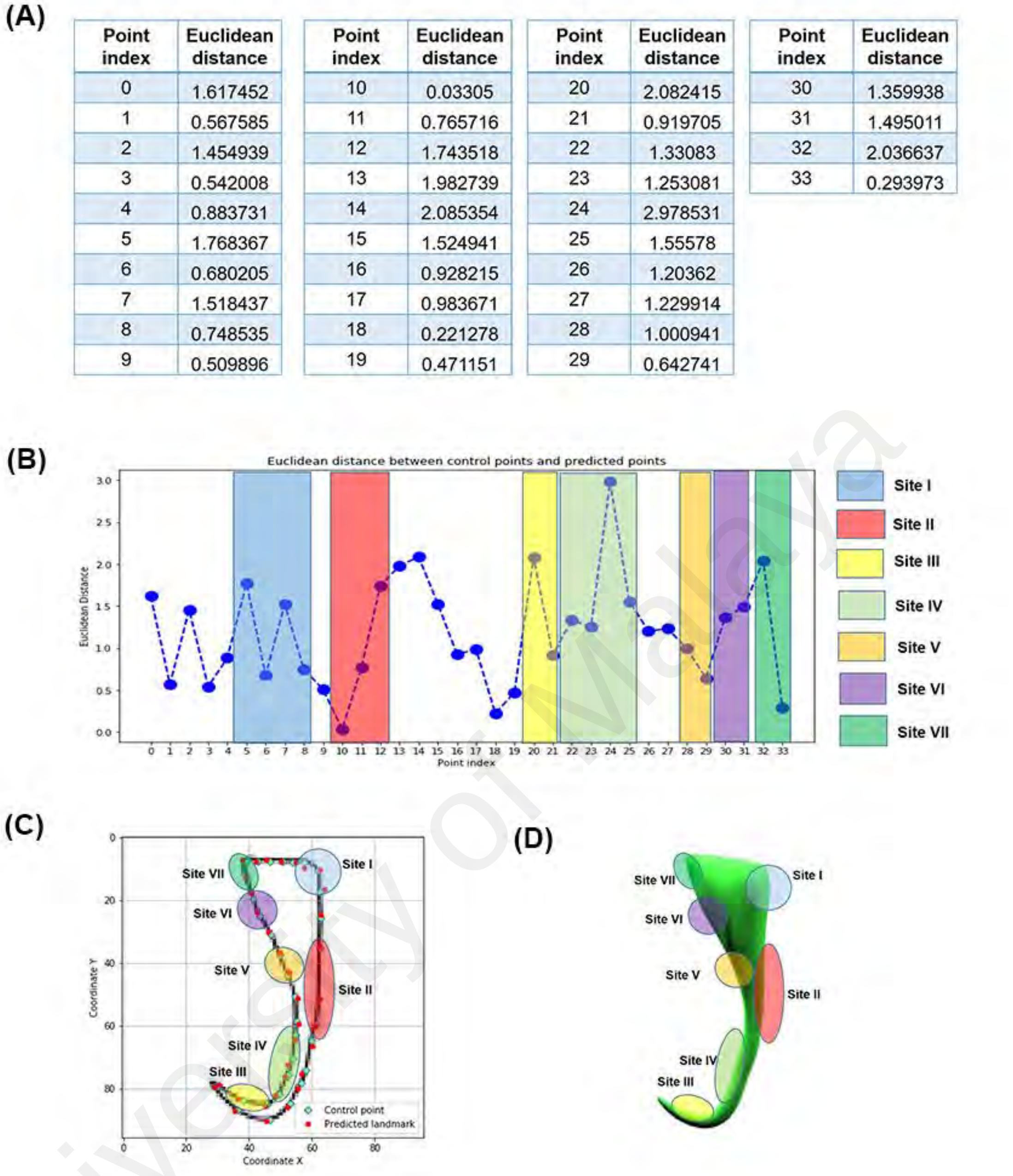
Figure 4.23: Quantitative analysis of shape accuracy of the derived 3D anchor (*Cichlidogyrus raeymaekersi*) based on Euclidean distance.

(A)



Point index	Euclidean distance	Point index	Euclidean distance	Point index	Euclidean distance	Point index	Euclidean distance
0	1.425105	10	1.080706	20	1.682093	30	2.555969
1	1.28318	11	1.902641	21	4.541941	31	0.802017
2	1.70253	12	1.937989	22	1.312786	32	0.280257
3	0.966867	13	1.355316	23	1.203992	33	0.498479
4	1.450916	14	2.072295	24	0.180319		
5	1.232888	15	4.047165	25	0.361516		
6	1.321225	16	4.573927	26	1.329444		
7	1.890195	17	1.226769	27	1.785892		
8	1.962525	18	2.641069	28	0.496535		
9	1.290833	19	1.011259	29	0.696818		

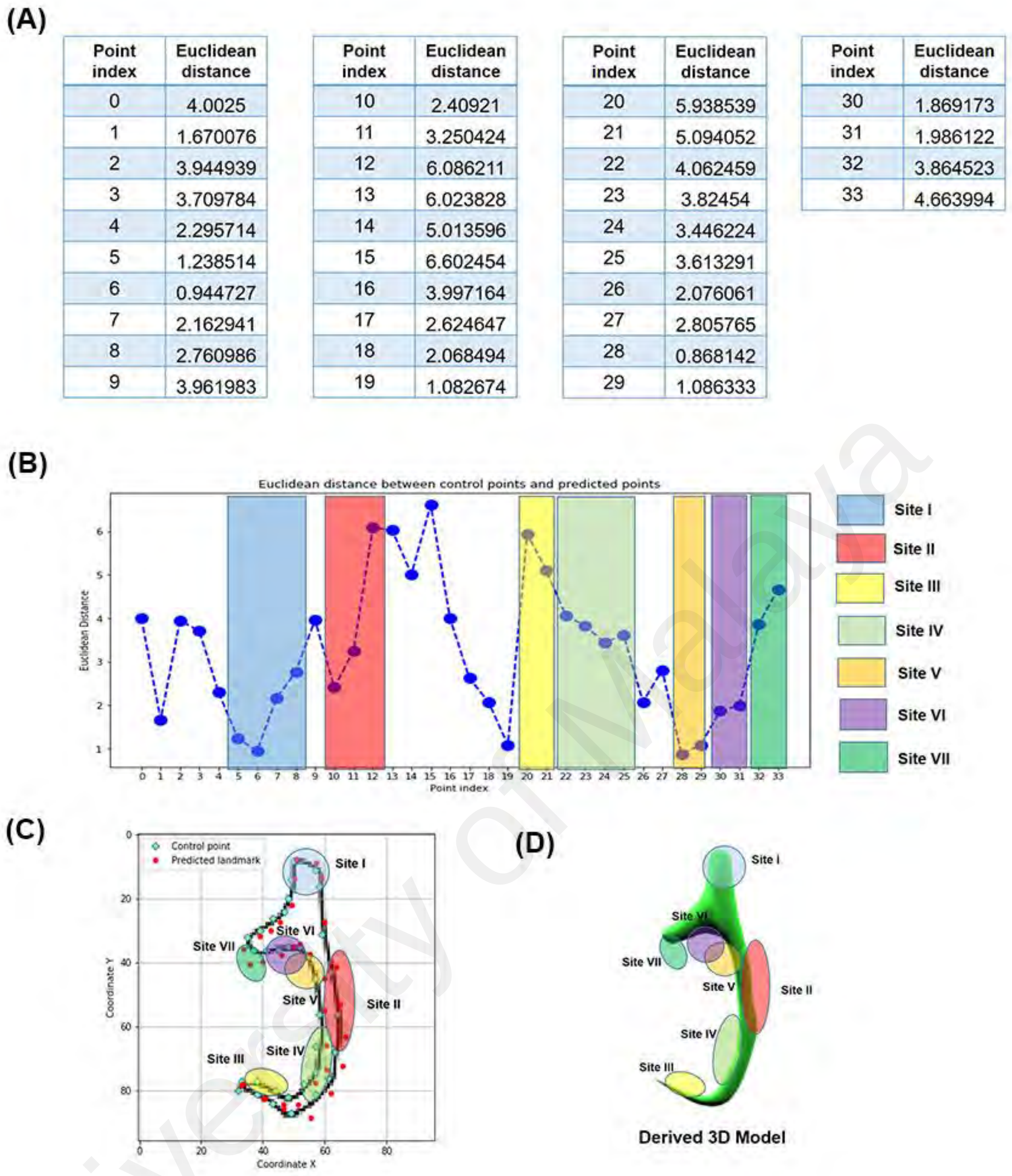
Figure 4.24: Quantitative analysis of shape accuracy of the derived 3D anchor (*Cichlidogyrus aspiralis*) based on Euclidean distance.



(A)

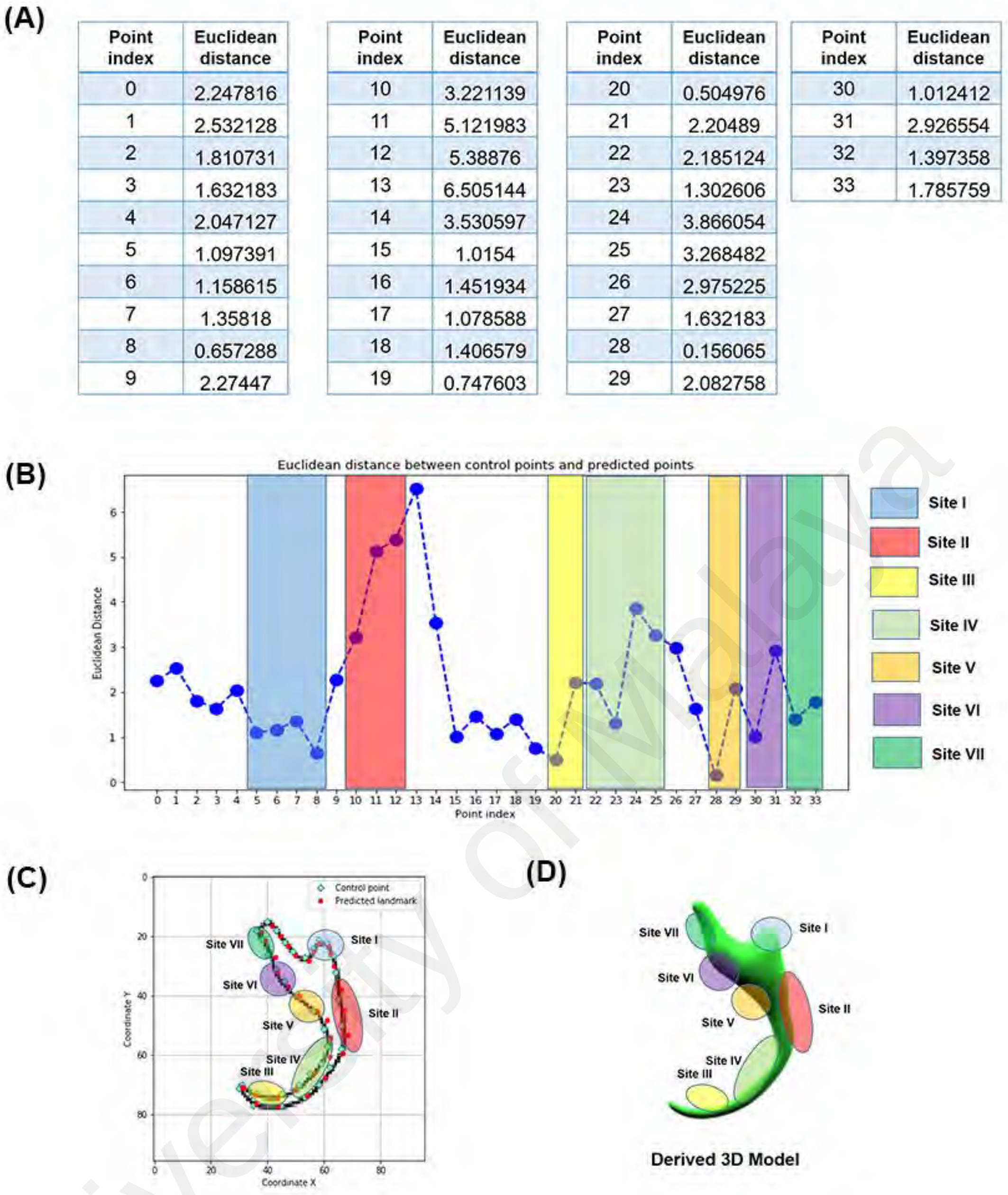
Point index	Euclidean distance	Point index	Euclidean distance	Point index	Euclidean distance	Point index	Euclidean distance
0	1.617452	10	0.03305	20	2.082415	30	1.359938
1	0.567585	11	0.765716	21	0.919705	31	1.495011
2	1.454939	12	1.743518	22	1.33083	32	2.036637
3	0.542008	13	1.982739	23	1.253081	33	0.293973
4	0.883731	14	2.085354	24	2.978531		
5	1.768367	15	1.524941	25	1.55578		
6	0.680205	16	0.928215	26	1.20362		
7	1.518437	17	0.983671	27	1.229914		
8	0.748535	18	0.221278	28	1.000941		
9	0.509896	19	0.471151	29	0.642741		

Figure 4.25: Quantitative analysis of shape accuracy of the derived 3D anchor (*Dactylogyrus primarus*) based on Euclidean distance.



Point index	Euclidean distance
0	4.0025
1	1.670076
2	3.944939
3	3.709784
4	2.295714
5	1.238514
6	0.944727
7	2.162941
8	2.760986
9	3.961983
10	2.40921
11	3.250424
12	6.086211
13	6.023828
14	5.013596
15	6.602454
16	3.997164
17	2.624647
18	2.068494
19	1.082674
20	5.938539
21	5.094052
22	4.062459
23	3.82454
24	3.446224
25	3.613291
26	2.076061
27	2.805765
28	0.868142
29	1.086333
30	1.869173
31	1.986122
32	3.864523
33	4.663994

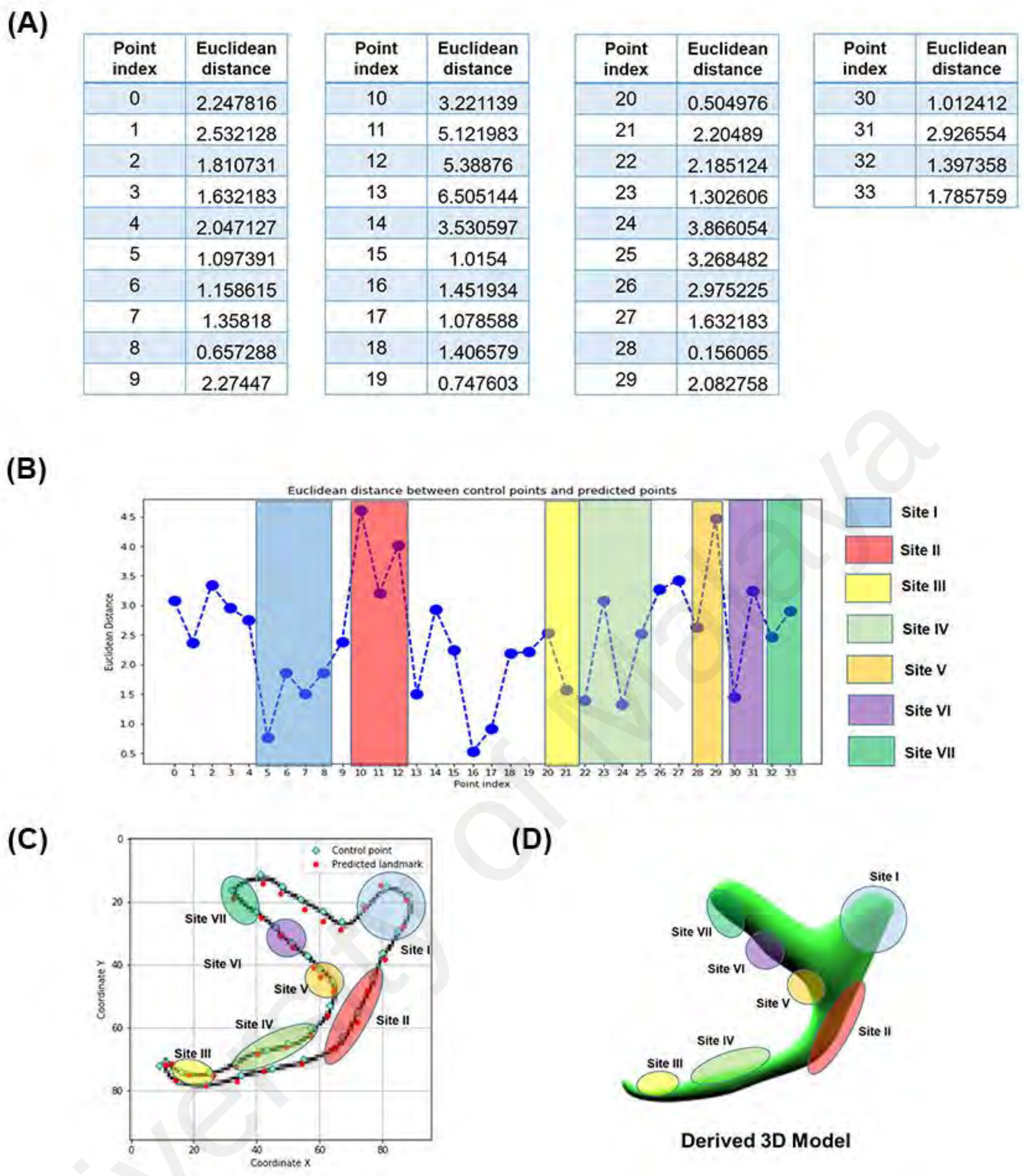
Figure 4.26: Quantitative analysis of shape accuracy of the derived 3D anchor (*Pellucidhaptor merus*) based on Euclidean distance.



(A)

Point index	Euclidean distance	Point index	Euclidean distance	Point index	Euclidean distance	Point index	Euclidean distance
0	2.247816	10	3.221139	20	0.504976	30	1.012412
1	2.532128	11	5.121983	21	2.20489	31	2.926554
2	1.810731	12	5.38876	22	2.185124	32	1.397358
3	1.632183	13	6.505144	23	1.302606	33	1.785759
4	2.047127	14	3.530597	24	3.866054		
5	1.097391	15	1.0154	25	3.268482		
6	1.158615	16	1.451934	26	2.975225		
7	1.35818	17	1.078588	27	1.632183		
8	0.657288	18	1.406579	28	0.156065		
9	2.27447	19	0.747603	29	2.082758		

Figure 4.27: Quantitative analysis of shape accuracy of the derived 3D anchor (*Dactylogyrus falcatus*) based on Euclidean distance.



(A)

Point index	Euclidean distance
0	2.247816
1	2.532128
2	1.810731
3	1.632183
4	2.047127
5	1.097391
6	1.158615
7	1.35818
8	0.657288
9	2.27447

Point index	Euclidean distance
10	3.221139
11	5.121983
12	5.38876
13	6.505144
14	3.530597
15	1.0154
16	1.451934
17	1.078588
18	1.406579
19	0.747603

Point index	Euclidean distance
20	0.504976
21	2.20489
22	2.185124
23	1.302606
24	3.866054
25	3.268482
26	2.975225
27	1.632183
28	0.156065
29	2.082758

Point index	Euclidean distance
30	1.012412
31	2.926554
32	1.397358
33	1.785759

Figure 4.28: Quantitative analysis of shape accuracy of the derived 3D anchor (*Dactylogyrus vastator*) based on Euclidean distance.

(A)

Point index	Euclidean distance	Point index	Euclidean distance	Point index	Euclidean distance	Point index	Euclidean distance
0	1.96939	10	2.423656	20	0.719405	30	1.636763
1	1.007821	11	3.199087	21	0.399193	31	0.812503
2	1.277828	12	1.57479	22	3.086505	32	1.911407
3	1.945429	13	2.684187	23	4.920418	33	1.784096
4	3.440041	14	1.103775	24	3.1571		
5	1.256611	15	1.251001	25	4.31985		
6	1.089169	16	3.698471	26	5.590568		
7	0.655281	17	2.166869	27	2.70747		
8	1.736396	18	2.71496	28	3.63712		
9	1.015793	19	1.50482	29	1.515986		

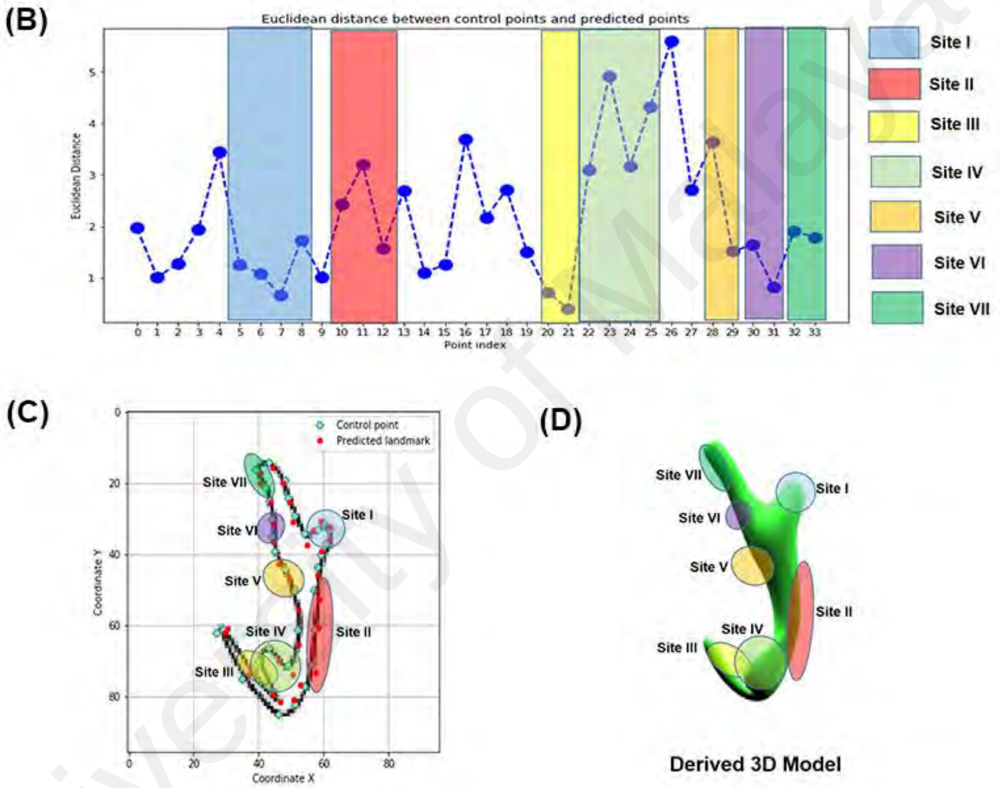


Figure 4.29: Quantitative analysis of shape accuracy of the derived 3D anchor (*Dactylogyrus pterocleidus*) based on Euclidean distance.

(A)

Point index	Euclidean distance	Point index	Euclidean distance	Point index	Euclidean distance	Point index	Euclidean distance
0	4.544522	10	4.832693	20	1.345441	30	3.190339
1	2.833342	11	6.496788	21	1.394202	31	3.063371
2	0.77935	12	4.512416	22	2.129194	32	3.730973
3	2.019374	13	1.134618	23	2.231636	33	2.543781
4	2.295978	14	0.698029	24	2.315446		
5	0.345972	15	1.366427	25	4.928649		
6	3.75958	16	2.130739	26	6.967202		
7	4.244346	17	1.551952	27	3.107036		
8	4.724079	18	3.4205	28	1.039988		
9	1.877973	19	3.395055	29	0.588354		

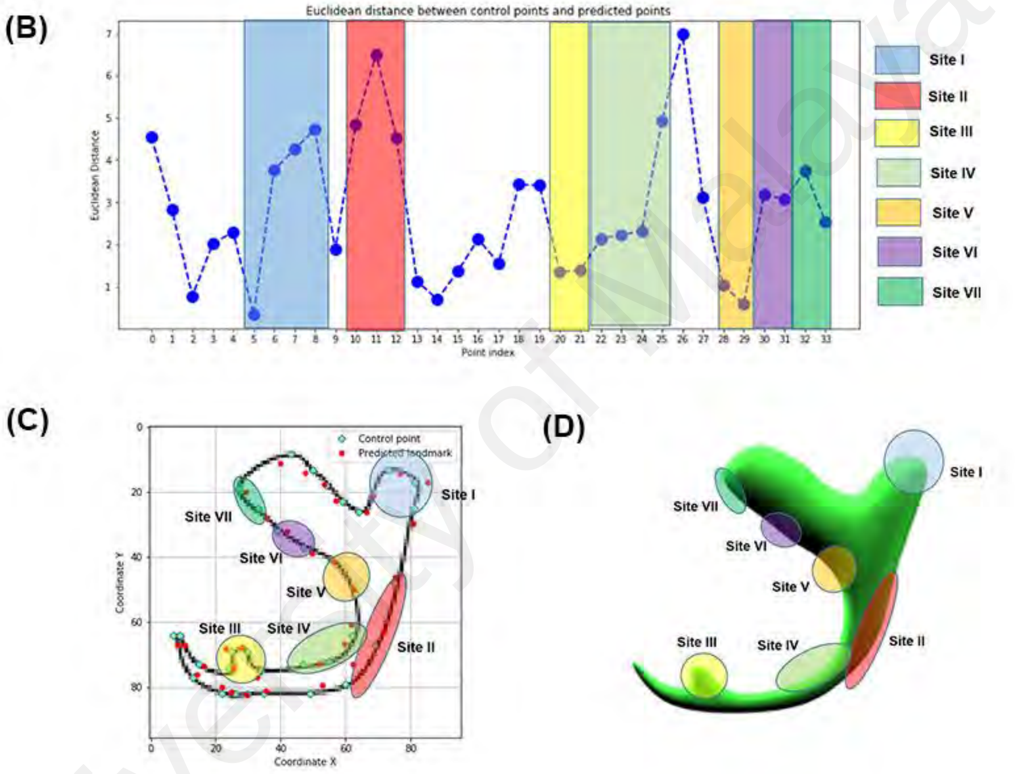
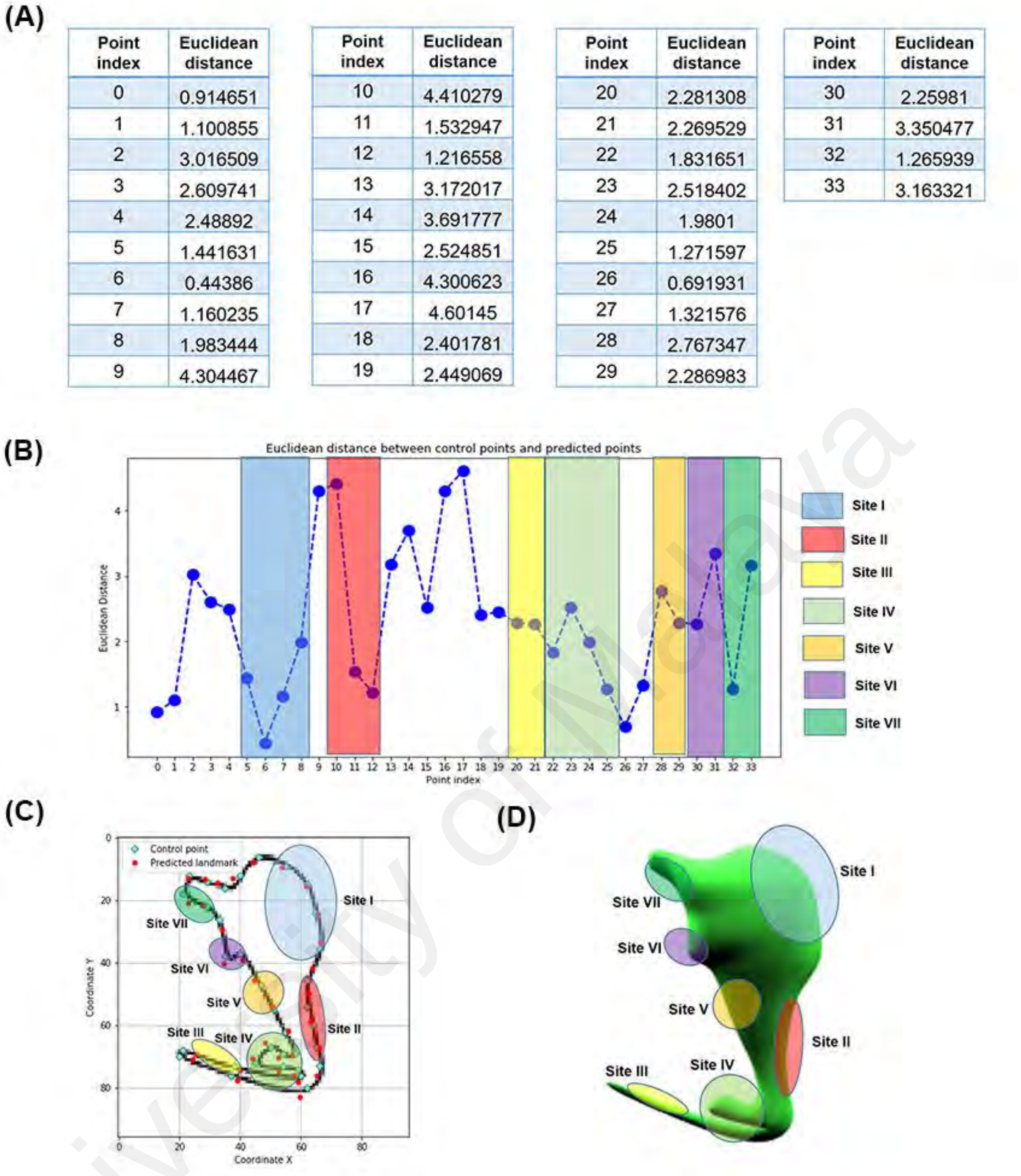
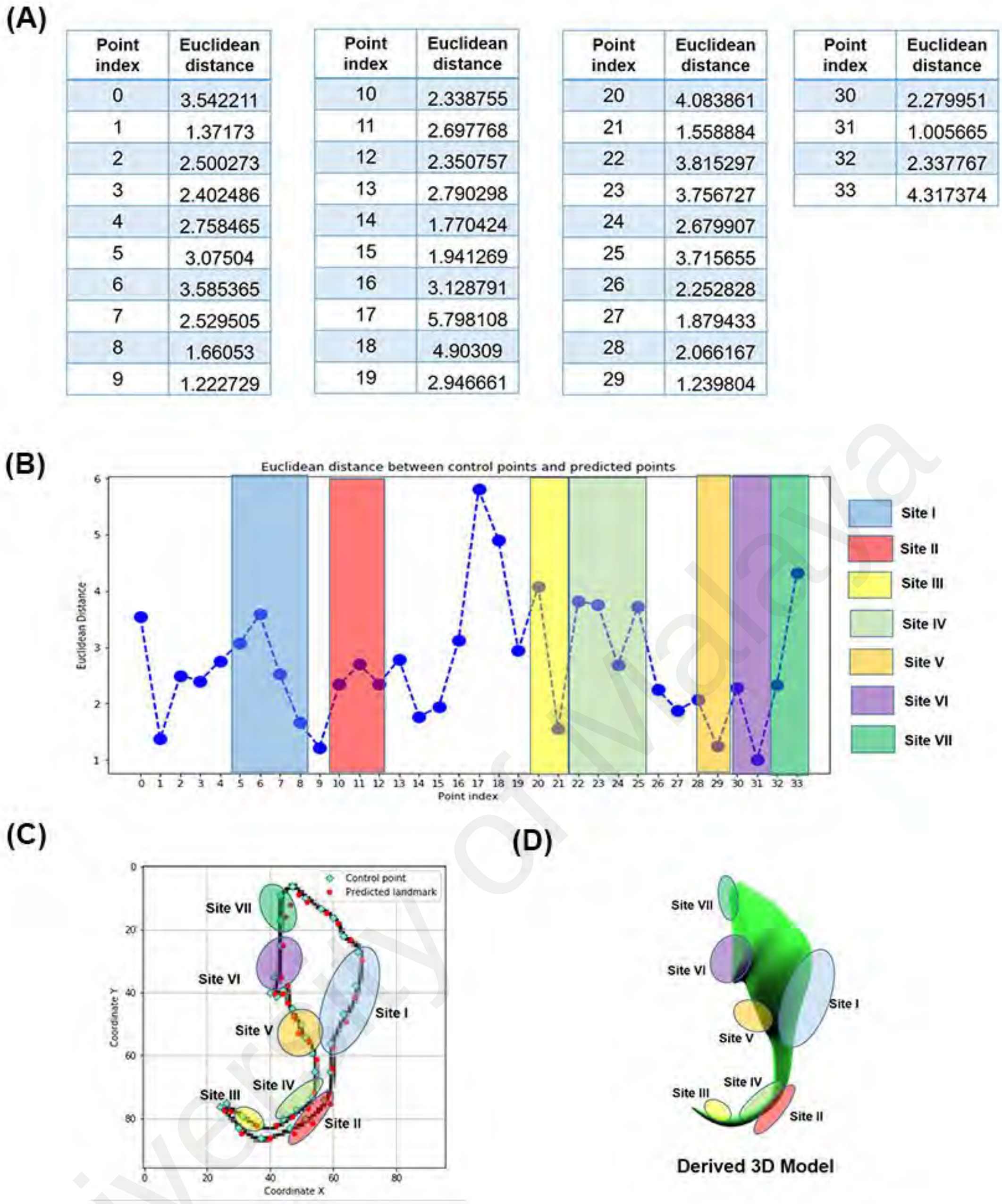


Figure 4.30: Quantitative analysis of shape accuracy of the derived 3D anchor (*Dactylogyrus falciunguis*) based on Euclidean distance.



Point index	Euclidean distance
0	0.914651
1	1.100855
2	3.016509
3	2.609741
4	2.48892
5	1.441631
6	0.44386
7	1.160235
8	1.983444
9	4.304467
10	4.410279
11	1.532947
12	1.216558
13	3.172017
14	3.691777
15	2.524851
16	4.300623
17	4.60145
18	2.401781
19	2.449069
20	2.281308
21	2.269529
22	1.831651
23	2.518402
24	1.9801
25	1.271597
26	0.691931
27	1.321576
28	2.767347
29	2.286983
30	2.25981
31	3.350477
32	1.265939
33	3.163321

Figure 4.31: Quantitative analysis of shape accuracy of the derived 3D anchor (*Chauhanellus auriculatum*) based on Euclidean distance.



Point index	Euclidean distance	Point index	Euclidean distance	Point index	Euclidean distance	Point index	Euclidean distance
0	3.542211	10	2.338755	20	4.083861	30	2.279951
1	1.37173	11	2.697768	21	1.558884	31	1.005665
2	2.500273	12	2.350757	22	3.815297	32	2.337767
3	2.402486	13	2.790298	23	3.756727	33	4.317374
4	2.758465	14	1.770424	24	2.679907		
5	3.07504	15	1.941269	25	3.715655		
6	3.585365	16	3.128791	26	2.252828		
7	2.529505	17	5.798108	27	1.879433		
8	1.66053	18	4.90309	28	2.066167		
9	1.222729	19	2.946661	29	1.239804		

Figure 4.32: Quantitative analysis of shape accuracy of the derived 3D anchor (*Chauhanellus caelatus*) based on Euclidean distance.

CHAPTER 5: DISCUSSION AND CONCLUSION

5.1 Introduction

In this chapter, each of the research questions presented in Chapter 1, Section 1.2 is addressed and discussed based on the research outcomes shown in Chapter 4. In addition to this, the research contributions are discussed, followed by the constraints and limitations observed in this study. Finally, the chapter is ended with a conclusion on the entire study and a discussion on future development work. The content flow of this chapter is as follows:

A. Feasibility of shape deformation of generic 3D model to derive 3D monogenean anchor
B. Effectiveness of using the data augmentation program developed in this study in generating training sets for machine learning
C. Feasibility of the NoSQL database to store training sets
D. Efficiency of using the automated 3D modelling pipeline empowered by the machine learning model to generate 3D models of the monogenean anchors
E. Research contributions
F. Constraints and limitation of this study
G. Conclusion
H. Future development

5.2 Feasibility of shape deformation of generic 3D model to derive 3D monogenean anchors

In this study, a generic 3D monogenean anchor model was successfully developed (Figure 4.1). This generic 3D model was first tested to derive the monogenean anchors of 8 selected species through the manual deformation approach. Although the structure of generic 3D model is simple, it can be deformed into the different required anchor shapes and forms using the manual deformation method (Figure 4.14) which proves the feasibility to adopt shape deformation approach to derive 3D models of the biological specimens such as the monogenean anchors from a generic 3D model. The core idea behind this manual deformation approach is to manually displace the point primitives of the generic 3D model to a new position in a Cartesian 3D space to fit the shape presented in an input 2D illustration of monogenean anchor. In this generic 3D model, all the constituent point primitives are used as pilot points during the deformation and this makes the model more deformable and versatile. Thus, a careful selection number of point primitives to build the generic 3D model is important to ensure that not too many or too few point primitives are assigned. This is crucial so that there are sufficient point primitives at sites of high morphological variations (Figure 3.2) to allow for deformation into the diverse forms and on another hand the deformation process would not be complicated by too many point primitives.

This study has also shown that a generic 3D model of monogenean anchor can be developed using a geometric modelling approach. Rectangular building blocks are used in constructing this deformable generic 3D model as indicated by the wireframes for the generic 3D model. The reason why the rectangular building blocks were used despite their angularities is because in principle such building blocks required the least number of point primitives. In addition, the rectangular building blocks can also be smoothened using a surface modifier such as Catmull-Clark subdivision modifier to obtain a more natural shape.

The manual shape deformation technique on a digital generic 3D model is proven to be feasible to derive another target 3D model of biological specimens like monogenean anchors. However this manual deformation approach is a tedious procedure as it demands a user to move the point primitives of the generic 3D model one after another to fit the shape of the input illustration. Hence, this impediment motivated the development of an automated 3D modelling pipeline which automates deformation process to derive target 3D models with minimal human intervention.

5.3 Feasibility of data augmentation generated training sets for machine learning

The automated 3D modelling pipeline developed in this study is empowered by a machine learning model to automate the deformation of the generic 3D model. A big data set of 2D illustrations of monogenean anchors were needed to train this machine learning model by the supervised learning procedure. The data augmentation program developed in this study has shown its capability to generate training sets for eight different morphological variant shapes of anchor. As aforementioned, a total of 5000 synthetic 2D illustrations were successfully generated using the program and the generation of such amount of illustrations could be a formidable task by manual hand drawing. The 2D shape interpolation algorithm embedded within this augmentation program can easily capture a sequence of interpolated 2D shapes as synthetic illustrations when a source shape is morphed into a target shape. Although some synthetic illustrations may show subtle difference from one another, their inherent pixel values and the coordinates of their constituent 2D coordinates of point primitives can show significant difference to qualify them as training set for supervised learning. In addition, the shape variation of the resulting synthetic 2D illustrations was also greatly increased by scaling and rotating the illustrations in different random degrees.

5.4 Feasibility of the NoSQL database to store training sets

In this study, all the 5000 synthetic illustration training sets are stored in a NoSQL database , MongoDB instead of a file system as in many other machine learning studies. One advantage of storing these training set in the database is that all the illustrations along with their associated labels such as the illustration name, shape category and coordinates of their associated point primitives are encapsulated into a consistent data model. Besides, unlike the rigid data schema as seen in a relational database such as MySQL, the data model of the NoSQL-based MongoDB database can be adjusted or extended without revamping the entire database system for future expansion. In contrast with the file system adopted by other machine learning studies which always store the image files and their associated labels in several separate text or CSV files, the MongoDB database offers a uniform storage platform to store all the required data. Such uniform storage platform enables a more systematic and robust data management compared with the traditional file system. A program module could easily be created by using the MongoDB API Interface to fetch the data from the database for the machine learning.

5.5 The usefulness of developed automated 3D modelling pipeline

The developed automated 3D modelling pipeline enables the entire process of deformation of the generic 3D model into a target model in a very direct and user friendly way. Compared with the manual deformation approach which manually align the point primitives of the generic 3D model one after another to fit the shape of the 2D template, the automated 3D modelling pipeline only requires a user input of 2D illustration and a single activation of the button to trigger the modelling system to automatically deform the generic 3D model into a 3D shape that resembles the input 2D illustration. Therefore the entire modeling process happens with minimal human intervention. This process is also less time consuming compared to the manual approach.

Besides, the versatility of the automated 3D modelling pipeline is dependent on the shape variability of the training sets catered to the machine learning model. This means the more 2D illustrations with additional morphological variants shapes used as the training sets, the more versatile the machine learning model to empower the 3D modelling pipeline to derive a more diverse 3D models. In this study, the automated 3D modelling pipeline has derived 3D models of the monogenean anchors from 12 monogenean species, which are *Cichlidogyrus dracolemma*, *Dactylogyrus wunderi*, *Cichlidogyrus raeymaekersi, Cichlidogyrus aspiralis, Dactylogyrus primaries, Pellucidhaptor merus, Dactylogyrus falcatus, Dactylogyrus vastator, Dactylogyrus pterocleidus, Dactylogyrus falciunguis, Chauhanellus auriculatum* and *Chauhanellus caelatus* (Figure 4.16 & Figure 4.17). The versatility of the automated 3D modelling pipeline can be expanded by catering more 2D illustrations with additional morphological variant shapes as training sets to train the machine learning model.

5.6 Research contributions

The entire project development of this study has shown several research contributions in the following aspects:

A. Shape deformation approach of 3D models construction for biological specimens

B. Data augmentation approach for machine learning

C. Design and construction of a NoSQL-based database to store image-type training sets

D. Application of a machine learning technique in 3D modelling of biological specimens
E. Usefulness of the 3D models of monogenean anchor in Biology

5.6.1 Shape deformation approach of 3D models construction for biological specimens

The shape deformation approach in this study to derive the 3D models of monogenean anchors from a generic 3D model is a novel way of 3D modelling without going through the tedious modelling process from scratch. A target 3D model can be derived by merely moving the point primitives of the generic model to deform it into a target 3D shape. This shape deformation approach is user friendly and obviates the need of adept modelling skills to create a 3D model. To the best of our knowledge, this shape deformation approach has not been applied in other relevant 3D modelling work to study morphology of microorganism like parasites while many researchers still rely on the usage of the costly 3D imaging modalities (Brooker et al., 2012; Galli et al., 2006; García-Vásquez et al., 2012; Hayes et al., 2011; Valverde-Islas et al., 2011). Besides, this shape deformation approach improves the 3D modelling work published in Teo et al. (2010), by obviating the usage of the comprehensive 3D modelling software, Autodesk 3ds Max to create a 3D model from scratch. This novel application of shape deformation to derive target 3D models of the monogenean anchors from a generic model was published in Teo et al. (2013).

5.6.2 Data augmentation for machine learning

A common approach to augment image training sets for a machine learning model is to apply transformations on an original image such as by rotating the image, flipping the image, randomly cropping the image, switching the color channels and adding noise

(Grigorev et al., 2017). However, such image transformations alone cannot cater the need of this study to generate synthetic 2D illustrations of anchor with diverse variations of shape. In this study, a 2D shape interpolation algorithm was developed and integrated in the data augmentation program to create the synthetic training sets for the machine learning. Without this 2D shape interpolation algorithm, it is practically infeasible to prepare the illustrations training sets that can capture a large variation of possible 2D shapes for even one type of hand drawn monogenean anchor. The data augmentation program supported by the 2D shape interpolation algorithm can effectively produce the synthetic illustrations with different level of morphological variant shapes in addition to applying random image transformation on them.

5.6.3 Development of a NoSQL database to store image-type training sets

With the emergence of big data, the practice of data storage has been shifted from traditional relational databases to NoSQL databases (Bâzâr & Sebastian, 2014). This study has echoed such trend of transition by adopting one variant of the NoSQL database, MongoDB as a storage system to store all the illustrations training sets along with their labelling details. While the MongoDB has been around in the market since 2009, there is still a paucity of published records on the application of MongoDB to store the image datasets for machine learning. This study has demonstrated an initiative to design a simple yet scalable data structure to encapsulate all the details of the image-type training sets (Figure 3.15). It is worthwhile to reiterate that all the stored illustrations are represented as an array of pixel values which is a highly portable data representation independent of any image format. In comparison to the Binary Large Object (BLOB) which is an image file datatype supported in a relational database such as MySQL (Metha et al., 2018), the pixels array is a more flexible data structure which can be directly deployed in machine learning without going through data conversion. This means the pixels array of every illustration training sets can be directly retrieved from the MongoDB and used as a raw data to train a machine learning model.

5.6.4 Application of a machine learning technique in 3D modelling of biological specimens

The automated 3D modelling pipeline developed in this study presents an innovative way to automate the deformation of the generic 3D model into a target 3D anchor by using a 2D illustration as an input. This entire modelling pipeline offers a very direct way to construct a target 3D model with minimal human intervention. The automation is achieved by having a machine learning model trained in the neural network to detect landmark points on the input 2D illustration. The point primitives of the generic 3D model are then automatically aligned with the predicted landmark points which will eventually ensue in deformation of the generic 3D model into the target 3D shape that fits the input 2D illustration. Such application of machine learning technique for landmark point detection is still not reported in any published records relevant to modelling biological specimens. Many machine learning studies for landmark point detection are still predominantly constrained in human facial analysis but not to create 3D models of biological specimens (Chen et al., 2017; Deng et al., 2018; Fan & Zhou, 2016; Gondhi et al., 2017; Ranjan et al., 2017; Zhang et al., 2016).

5.6.5 Usefulness of the 3D models of monogenean anchor in Biology

Although biological organisms are three dimensional in nature, their images (line drawings and photographs) particularly in taxonomy are two dimensional and thus provide an incomplete visualization of their structural morphology. Rotations of the 3D models have shown that variations in shape could be caused by the orientation of the organism during preservation as indicated by the different views of anchor generated by rotating the 3D anchors through 360° (Figure 4.15). This shows that the way the diagnostic parts are oriented during collection and preservation process can affect the shape and morphometric measurements of the specimens and subsequently their taxonomic identification. This supports the need to flatten the worms in such a way as to have a horizontally disposed anchor (lateral view), which gives the best view for measurement and taxonomic comparison (Lim & Gibson, 2009). There are instances where specimens belonging to the same species are considered different based on variations in anchor shape caused by orientation. By comparing the shapes of a 3D anchor of a monogenean species in different rotational orientation researchers can determine whether the anchors belong to the same or different species. It should also be noted that the 3D models have the advantage of being more versatile and the researchers can interactively manipulate the constructed 3D shape for a better understanding of how the different structures function.

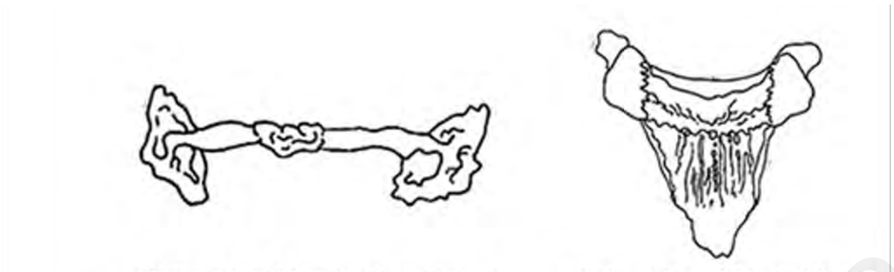
5.7 Constraints and limitations

There are some constraints and limitations found in the current study which are discussed based on several aspects as follows:

A. Rectangular building blocks in generic 3D model
B. Database setup
C. Limitations of the automated 3D modelling pipeline

5.7.1 Rectangular building blocks in generic 3D model

Although the usage of the rectangular building blocks is viable to construct a presentable generic 3D model for a monogenean anchor (Figure 4.1), it might not be the case for other sclerite structures of monogenean (Figure 5.1) which are much more complex and intricate structure. This means the application of the similar geometric modelling procedure (Refer Chapter 3, Section 3.2) is not applicable to create a generic 3D model of other sclerite structure to derive another target group of 3D shapes which have more complicated geometry.



(A) Dorsal bar of *Gyrodactylus gasterostei* **(B)** Ventral bar of *Gyrodactylus gasterostei*

(C) Haptoral terminal papillar sclerite of *Dendromonocotyle colorni* **(D)** Haptoral septal sclerite of *Dendromonocotyle colorni* **(E)** Septal sclerite of *Dendromonocotyle colorni*

Figure 5.1: Sclerite structures with intricate surfaces. Adapted from Chisholm et al. (2001) and Shinn et al. (1993).

Nevertheless, the construction of the generic 3D model in this study has demonstrated a novel idea of using geometric modelling method to build a generic model without being dependent on complex modelling algorithms or expensive 3D scanning hardware. A possible solution to obtain a more natural shape for such complicated biological structure is to combine primitive shape such as cylinder and sphere into a single 3D geometry instead of using only rectangular building block (Karčiauskas & Peter, 2012).

5.7.2 Database setup

The MongoDB database developed in this study was set up in a local machine environment. While the database offers a consistent and uniform data management of the training sets, the size of the storage is still dependent on the capacity of the local machine. Such dependency on the capacity of the local machine will restrict the scalability of the database to store a growing training set. Besides, the transaction speed of the database to query the data for the machine learning is restricted by the processing power of the local machine. Since the dataset in this study is expected to be big, latency will be observed during runtime. In short, the features offered by MongoDB such as data scalability and optimized processing performance are compromised and constrained by the inherent storage capacity and processing power of the local machine. One viable solution to resolve this issue is to migrate the database to a cloud server (See Section 5.8 below).

5.7.3 Limitations of the automated 3D modelling pipeline

As highlighted in the Figure 4.18, the automated 3D modelling pipeline does not work well on deriving the 3D models of anchors with some intricate structures such as the extrusions found in the species of *D. pterocleidus* and *C. auriculatum* and the hook in *C. caelatus.* Such distortion of shape is due to the fact that these morphological structures (e.g. extrusions and hook) possess a relatively smaller surface area to be controlled by a higher number of point primitives compared with other regions of the anchor. This means a slight diversion of the predicted coordinates of those point primitives from the control points will cause an obvious distortion of shape.

Besides, the automated 3D modelling pipeline can only work on the input 2D illustration with grayscale color and with a fixed size of 96 x 96 pixels. The shape in the input 2D illustration must also be laterally positioned. This is because the machine learning model that empowers the 3D modelling pipeline was trained with the synthetic 2D illustrations of anchor which are laterally positioned, and all the sizes and colors are standardized to 96x96 pixels and grayscale level. Hence, the machine learning model cannot effectively detect the landmark points on a wild image for auto-deformation of the generic 3D model to derive a target 3D shape. One must preprocess the input 2D illustration before it can be fetched into the automated 3D modelling pipeline to derive a target 3D model.

5.8 Conclusion

In this study, an automated 3D modelling pipeline that uses 2D illustration as an input to generate a variety of digital 3D models was developed as shown in Figure 4.16 and Figure 4.17. This automated 3D modelling pipeline is backed by a machine learning algorithm that can detect the landmark points' location on an input 2D illustration and automatically align them with the vertices of a generic 3D model to enable deformation of its 3D shape to produce a target 3D model.

In the methodology, a generic 3D model of the monogenean anchor was developed which can be deformed into diverse forms of monogenean anchors via both the manual and automated deformation approach. Besides, a data augmentation technique which managed to capture a sequence of 2D interpolated shapes as synthetic 2D illustrations was developed. A total of 5000 synthetic illustrations were generated as training set and were stored in a NoSQL database developed using MongoDB. A machine learning model was also trained via a supervised learning process using the 5000 synthetic illustrations retrieved from the NoSQL database. The machine learning model was then integrated into the automated 3D modelling pipeline and it managed to derive the 3D models of monogenean anchor which are generally comparable to the ones derived by manual deformation approach (Figure 4.19). Instead of manually moving the point primitives one after another to deform the generic 3D model into a target shape, the automated 3D modelling pipeline empowered by the machine learning model has automated the deformation process to obtain a desired 3D shape with a single user input illustration and a single activation of button.

Besides, the Euclidean distance analysis as presented in Figure 4.21 – 4.32 and the information presented in Table 4.3 have shown most of the identified sites of high morphological variations in the 3D models produced by the automated 3D modelling pipeline are tolerable to the shape distortion. The slight distortions of the shape (e.g. slight bend up or down) would not affect the overall visualization of the resulting 3D models. If the predicted landmark points are still positioned on the edge of the 2D illustrations, the shape distortion is almost unnoticeable even though the Euclidean distance between the control points and predicted points are relatively high. Hence, this Euclidean distance analysis verifies that the machine learning model is capable of predicting landmark points which can be used to derive a plausible 3D model of monogenean anchor. Nevertheless, an obvious shape distortion was observed on the hook and extrusion of some anchors. This shows that the shape distortion is greatly impacted by even a slight diversion of the predicted landmark points from the control points especially on those sites of high morphological variations with much lower surface area such as the extrusion and hook of some anchors.

Despite some constraints and limitation as discussed above, the automated 3D modelling pipeline developed in this study has demonstrated a working idea of the application of machine learning approach in a 3D modelling work. The automated 3D modelling pipeline developed in this study has been successfully used to derive 12 selected monogenean anchors from the generic 3D model and their shapes were shown to resemble to the 2D outline as presented in each of their 2D illustration template although there are some flaws observed on some anchors.

5.9 Future development

The automated 3D modelling pipeline developed in this study is a prototype to show the application of machine learning technique to automate the deformation of a generic 3D model to derive another target 3D model by using a 2D illustration as an input. In this study, only the monogenean anchor from 12 selected species were used as the testing objects to verify the usefulness of the automated 3D modelling pipeline in deriving the target 3D shapes from the generic 3D anchor. This entire 3D modelling pipeline is also applicable to create 3D models for different hard parts of monogeneans such as marginal hook, bars, clamps and suckers and male and female copulatory organs as well as hooks of trypanorhynch, other cestodes and acanthocephalans and any other organisms. As part of the future development plan, a generic 3D model will be created for each of the monogenean hard parts respectively and the training sets preparation and machine learning process will be repeated to train a machine learning model which is more versatile than the current model to detect landmark points on the input 2D illustrations of different monogenean hard parts. These predicted landmark points are expected to go through the same automated 3D modelling pipeline to derive the target 3D shapes from their respective generic 3D model. Once the 3D models of other sclerotized hard parts of monogenean are obtained, they can be assembled into a 3D body outline of a monogenean.

In order to achieve the future development plan, further studies on geometric modelling methods are required to investigate the feasibility to utilize a combination of primitive shapes such as cylinder and sphere to build generic 3D models for different target groups of monogenean structure as mentioned above. Besides, the entire project development will be migrated to a cloud computing platform. This is because much more illustration training sets are needed to train the machine learning model, and this may be constrained by the limited storage capacity and the processing power of the existing local machine environment as discussed above. The cloud computing platform can be harnessed to store the escalating number of training sets in a scalable cloud server and at the same time the powerful computing engine of cloud platform can be utilized to build and deploy the machine learning model for the automated 3D modelling pipeline (Wright, 2018). In the long term, migration to the cloud platform is essential move to scale up the development of the automated 3D modeling pipeline presented in this study.

REFERENCES

Aifanti, N., Papachristou, C., & Delopoulos, A. (2010). The MUG facial expression database. *Proceedings of the 11th International Workshop on Image Analysis for Multimedia Interactive Services*, 1-4.

Ansari, A-N., & Abdel-Mottaleb, M. (2003). 3D face modeling using two orthogonal views and a generic face model. *Proceedings of the International Conference on Multimedia and Expo. 3*, III-289.

Antonelli, L., Quilichini, Y., & Marchand, B. (2010). Biological study of *Furnestinia echeneis* Euzet and Audouin 1959 (Monogenea: Monopisthocotylea: Diplectanidae), parasite of cultured Gilthead sea bream *Sparus aurata* (Linnaeus 1758) (Pisces: Teleostei) from Corsica. *Aquaculture*, *307*, 179-186.

Baltrušaitis, T., Robinson, P., & Morency, L.P. (2016). OpenFace: An open source facial behavior analysis toolkit. *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1-10.

Baran, I., & Popović, J. (2007). Automatic rigging and animation of 3D characters. *ACM Transactions on Graphics*, *26* (3), 72.

Bates, K.T., Manning, P.L., Hodgetts, D., & Sellers, W.I. (2009). Estimating mass properties of dinosaurs using laser imaging and 3D computer modelling. *PLoS ONE*, *4*(2), e4532.

Băzăr, C., & Sebastian, C. (2014). The transition from RDBMS to NoSQL. A comparative analysis of three popular non-relational solutions: Cassandra, MongoDB and Couchbase. *Database Systems Journal*, *5*(2), 49-59.

Belhumeur, P.N., Jacobs, D.W., Kriegman, D.J., & Kumar, N. (2013). Localizing parts of faces using a consensus of exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(12), 2930-2940.

Berry, R.P., & Ibbotson, M.R. (2010). A three-dimensional atlas of the honeybee neck. *PLoS ONE*, *5*(5), e10771.

Blanco, F.R., & Oliveira, M.M. (2008). Instant mesh deformation. *Proceedings of the Symposium on Interactive 3D Graphics and Games*, 71-78.

Bonaccorso, G. (2017). *Machine learning algorithms*. Birmingham, UK: Packt Publishing Ltd.

Bromiley, P.A., Schunke, A.C., Ragheb, H., Thacker, N.A., & Tautz, D. (2014). Semi-automatic landmark point annotation for geometric morphometrics. *Frontiers in Zoology*, *11*(1), 61.

Brooker, A.J., Shinn, A.P., & Bron, J.E. (2012). Use of laser scanning confocal microscopy for morphological taxonomy and the potential for digital type specimens (e-types). *Aquatic Biology*, *14*, 165-173.

Cevikalp, H., Triggs, B., & France, V. (2013). Face and landmark detection by using cascade of classifiers. *Proceedings of the 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, 1-7.

Chen, Y, Yang, J., & Qian, J. (2017). Recurrent neural network for facial landmark detection. *Neurocomputing*, *219*, 26-38.

Cheutet, V., Catalano, C.E., Pernot, J.P., Falcidieno, B., Giannini, F., & Leon, J.C. (2005). 3D sketching for aesthetic design using fully free-form deformation features. *Computer & Graphics*, *29*(6), 916-930.

Chollet, F. (2017). *Deep learning with Python.* New York, USA: Manning Publications Co.

Chisholm, L.A., Kearn, G., & Whittington, I.D. (2001). *Dendromonocotyle colorni* sp.n. (Monogenea: Monocotylidae) from the skin of *Himantura uarnak* (Dasyatididae) from Israel and a new host record for *D.octodiscus* from the Bahamas. *Folia Parasitologica*, *48*(1), 15-20.

Coates, A., Lee, H., & Ng, A.Y. (2011). An analysis of single-layer networks in unsupervised feature learning. *Journal of Machine Learning Research*, *15*, 215-223.

Cortes, C., & Vapnik, V. (1995). Support vector network. *Machine Learning*, *20*, 273-297.

Criminisi, A., Shotton, J., Robertson, D., & Konukoglu, E. (2011). Regression forests for efficient anatomy detection and localization in CT Studies. In Menze, B., Langs, G., Tu, Z., & Criminisi, A. (Eds.) *Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging. MCV 2010. Lecture Notes in Computer Science*, (pp. 106-117). Berlin, Heidelberg: Springer.

Dantone, M., Gall, J., Fanelli, G., & Gool, V.L. (2012). Real-time facial feature detection using conditional regression forests. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2578-2585.

Delp, S.L., Anderson, F.C., Arnold, A.S., Loan, P., Habib, A., John, C.T., … Thelen, D.G. (2007). OpenSim: Open-source Software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering*, *54* (11), 1940-1950.

Deng, W., Fang, Y., Xu, Z., & Hu, J. (2018). Facial landmark localization by enhanced convolutional neural network. *Neurocomputing*, *273*, 222-229.

Dhillon, S.K. (2018). Biological databases. In Ranganathan, S., Gribskov, M, Nakai, K. & Schönbach, C. (Ed.), *Encyclopedia of Bioinformatics and Computational Biology, vol. 2*, (pp. 96–117). Oxford: Elsevier.

du Preez, L.H., & Lim, L.H.S. (2000). *Neopolystoma liewi* sp. n. (Monogenea: Polystomatidae) from the eye of the Malayan box turtle (*Cuora amboinensis*). *Folia Parasitology*, *47*, 11–16.

du Preez, L.H., & Moeng, I.A. (2003). Additional morphological information on *Oculotrema hippopotami* Stunkard, 1924 (Monogenea: Polystomatidae) parasitic on the African hippopotamus. *African Zoology*, *39*, 225–233.

Ercan, M.Z. (2014). Evaluation of NoSQL databases for EHR systems. *Proceedings of the 25th Australasian Conference on Information Systems*, 1–10.

Erolin, C., Jarron, M., & Csetenyi, L.J. (2017). Zoology 3D: Creating a digital collection of specimens from the D'Arcy Thompson Zoology Museum. *Digital Applications in Archaeology and Cultural Heritage*, *7*, 51-55.

Fan, H., & Zhou, E. (2016). Approaching human level facial landmark localization by deep learning. *Image and Vision Computing*, *47*, 27-35.

Fannes, W., Vanhove, M.P.M, & Huyse, T. (2017). Redescription of *Cichlidogyrus tiberianus* Paperna, 1960 and *C. dossoui* Douëllou, 1993 (Monogenea: Ancyrocephalidae), with special reference to the male copulatory organ. *Systematic Parasitology*, *94*, 133-144.

Farfade, S.S., Saberian, M.J., & Li, L.J. (2015). Multi-view face detection using deep convolutional neural networks. *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, 643-650.

Ferrari, C., Lisanti, G., & Berretti, S. (2017). A dictionary learning-based 3D morphable shape model. *IEEE Transactions on Multimedia*, *19* (12), 2666-2679.

Galli, P., Strona, G., Villa, A.M., Benzoni, F., Doglia, S.M., & Kritsky, D.C. (2006). Three-dimensional imaging of monogenoidean sclerites by laser scanning confocal fluorescence microscopy. *Journal of Parasitology*, *92*(2), 395-399.

García-Vásquez, A., Shinn, A.P. & Bron, J.E. (2012). Development of a light microscopy stain for the sclerits of *Gyrodactylus* von Nordmann, 1832 (Monogenea) and related genera. *Parasitology Research*, *110*(5), 1639-1648.

Géron, A. (2017). *Hands-on machine learning with Scikit-Learn and TensorFlow.* CA, USA: O'Reilly Media.

Geurts, P., Irrthum, A., & Wehenkel, L. (2009). Supervised learning with decision tree-based methods in computational and systems biology. *Molecular BioSystems*, *5*(12), 1593-1605.

Gondhi, N.K., Kour, N., Effendi, S., & Kaushik, K. (2017). An efficient algorithm for facial landmark detection using haar-like features coupled with corner detection following anthropometric constraints. *Proceedings of the 2nd International Conference on Telecommunication and Networks (TEL-NET)*, 1-6.

Gross, R., Matthews, I., Cohn, J., Kanade, T., & Baker, S. (2010). Multi-PIE. *Image and Vision Computing*, *28*(5), 807-813.

Gusev, A.V. (1976). Freshwater Indian Monogenoidea. Principles of systematic, analysis of the world faunas and their evolution. *Indian Journal of Helminthology, 25 & 26*, 1–241.

Hadied, M.A., Khidr, A.A., Samak, O.A., & Said, A. (2015). Surface topography of the anterior adhesive apparatus of the gill monogenean parasite *Diplectanum* sp. Diesing, 1858, with some surface criteria. *Fisheries and Aquaculture Journal*, *6*(4), 150.

Hähnel M., Wiratanaya A., & Kraiss K.F. (2006). Facial expression modelling from still images using a single generic 3D head model. In Franke K., Müller KR., Nickolay B., Schäfer R. (Eds.), *Pattern Recognition. DAGM 2006. Lecture Notes in Computer Science*, (pp. 324-333). Springer, Berlin: Heidelberg.

Han, J., & Moraga, C. (1995). The influence of the sigmoid function parameters on the speed of backpropagation learning. In Mira, J. & Sandoval, F. (Eds.) *From Natural to Artificial Neural Computation. IWANN 1995. Lecture Notes in Computer Science*, (pp. 195-201). Berlin, Heidelberg: Springer.

Hayes, P.M., Wertheim, D.F., Smit, N.J., Seddon, A.M., & Davies, A.J. (2011). Three-dimensional visualization of developmental stages of an apicomplexan fish blood parasite in its invertebrate host. *Parasites & Vectors*, *4*, 219.

Hodová, I., Sonneck, R., & Geinar, M. (2018). Architecture of *Paradiplozoon homoiion*: A diplozoid monogenean exhibiting highly-developed equipment for extoparasitism. *PLoS ONE*, *13*(2), e0192285.

Ho, K.T. (1995). Random decision forests. *Proceedings of the Third International Conference on Document Analysis and Recognition*, *1*, 278-282.

Huang. H., Kalogerakis, E., Yumer, E., & Mech, R. (2017). Shape synthesis from sketches via procedural models and convolutional networks. *IEEE Transactions on Visualization and Computer Graphics*, *23*(8), 2003-2013.

Huang, F., Sun, T., & Bu, F. (2016). Generation of person-specific 3D model based on single photograph. *Proceedings of the 2nd IEEE International Conference on Computer and Communications (ICCC)*, 704-707.

Isaacson, C. (2014). *Understanding big data scalability: Big data scalability series, Part 1*. Crawfordsville, Indiana: Prentice Hall.

Jang, D-K., & Lee, S-H. (2017). Regression-based locating landmark on dynamic humans. *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 1-2.

Jeong, M., Nam, J-Y., & Ko B.C. (2017). Eye pupil detection system using an ensemble of regression forest and fast radial symmetry transform with a near infrared camera. *Infrared Physics & Technology*, *85*, 44-51.

Jia, S., & Cristianini, N. (2015). Learning to classify gender from four million images. *Pattern Recognition Letters*, *58*, 35-41.

Jia, X., Yang, H., Lin, A., Chan, K.P., & Patras, I. (2014). Structured semi-supervised forest for facial landmarks localization with face mask reasoning. *Proceedings of British Machines Visualization Conference (BMVA)*, doi:10.5244/C.28.85.

Kamsin, A., Teng, C.G., & Tay, L.K. (2007). The Implementation of a web-based 3D multimedia learning prototype in biology subject: BioID. *Proceedings of the 2007 Innovations in Information Technologies (IIT),* 133-137.

Kass, M., Witkin. A., & Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, *1*, 312–333.

Karčiauskas, K., & Peters, J. (2012). Free-form splines combining NURBS and basic shapes. *Graphical Models*, *74*, 351–360.

Kerbl, A., Handschuh, S., Nödl, M.T., Metscher, B., Walzl, M., & Wanninger, A. (2013). Micro-CT in cephalopod research: Investigating the internal anatomy of a sepiolid squid using a non-destructive technique with special focus on the ganglionic system. *Journal of Experimental Marine Biology and Ecology*, *447*, 140-148.

Khalid, S., Khalil, T., & Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. *Proceedings of the Science and Information Conference*, 372-378.

Khang, T.F., Soo, O.Y.M, Tan, W.B., & Lim. L.H.S. (2016). Monogenean anchor morphometry: Systematic value, phylogenetic signal and evolution. *PeerJ*, *4*, e1668.

Kho, Y., & Garland, M. (2007). Sketching mesh deformations. *Proceedings of the ACM SIGGARPH 2007 courses*, 41.

Khorrami, P., Paine, T.L., & Huang, T.S. (2015). Do deep neural network learn facial action units when doing expression recognition? *Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 19-27.

Köstinger, M., Wohlhart, P., Roth, P.M., & Bischof, H. (2011). Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. *Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2144-2151.

Kremer, A., Lippens, S., Bartunkova, S., Asselbergh, B., Blanpain, C., Fendrych, M., … Guerin, C.J. (2015). Developing 3D SEM in a broad biological context. *Journal of Microscopy*, *259*(2), 80-96.

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems, 1*, 1097-1105.

Laforsch, C., Imhof, H., Sigl, R., Settles, M., Heß, M., Wanninger, A. (2012). Applications of computational 3D–modeling in Organismal Biology. In Alexandru C. (Eds.), *Modeling and Simulation in Engineering* (pp. 117-142). Rijeka: InTech.

Le, B.H., & Deng, Z. (2017). Interactive cage generation for mesh deformation. *Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 1-9.

Le, V., Brandt, J., Lin, Z., Bourdev, L., & Huang, T.S. (2012). Interactive facial feature localization. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (Eds.) *Computer Vision – ECCV 2012. ECCV 2012. Lecture Notes in Computer Science*, (pp. 679-692). Springer: Berlin, Heidelberg.

Lee, C.H., Blackband, S.J., & Fernandez-Funez, P. (2015). Visualization of synaptic domains in the Drosophila brain by magnetic resonance microscopy at 10 micron isotropic resolution. *Scientific Reports*, *5*, 8920.

Lee, K.S.,Wong, K.H., Or, S.h., & Fung Y.F. (2001). 3D Face modeling from perspective views and contour-based generic-model. *Real-Time Imaging*, *7*, 173-182.

Lee, S.W.Y., & Tsai, C.C. (2013). Technology-supported learning in secondary and undergraduate biological education: Observations from literature review. *Journal of Science Education and Technology*, *22*(2), 226-233.

Le-Tien, T., & Pham-Chi, H. (2014). An approach for efficient detection of cephalometric landmarks. *Procedia Computer Science*, *37*, 293-300.

Li, Y., & Manoharan, S. (2013). A performance comparison of SQL and NoSQL databases. *Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, 15-19.

Lim, L.H.S. (1986). New species of *Trianchoratus* Price et Berry, 1966 (Ancyrocephalidae) from Malayan anabantoid fishes. *Parasitologia Hungarica, 19*, 13-42.

Lim, L.H.S. (1994). *Chauhanellus* Bychowsky and Nagibina, 1969 (Monogenea) from ariid fishes (Siluformes) of Peninsular Malaysia, *Systematic Parasitology*, *28*, 99-124.

Lim, L.H.S. (1995). *Bravohollisia* Bychowsky and Nagibina, 1970 and *Caballeria* Bychowsky and Nagibina, 1970 (Monogenea, Ancyrocephalidae) from *Pomadasys hasta* (Bloch) (Pomadasyidae), with the description of a new attachment mechanism. *Systematic Parasitology*, *32*, 211–224.

Lim, L.H.S. (1995). *Neocalceostoma* Tripathi, 1957 and *Neocalceostomoides* Kritsky, Mizelle & Bilqees, 1978 (Monogenea: Neocalceostomatidae n. fam.) from ariid fishes of Peninsular Malaysia. *Systematic Parasitology*, *30*, 141–151.

Lim, L.H.S. (1996). Eight new species of *Hamatopeduncularia* Yamaguti, 1953 (Monogenea: Ancyrocephalidae) from Arridae of Peninsular Malaysia. *Systematic Parasitology*, *33*, 53–71.

Lim, L.H.S. (1998). Diversity of monogeneans in Southeast Asia. *International Journal of Parasitology*, *28*, 1495-1515.

Lim, L.H.S., & du Preez, L.H. (2001). *Sundapolystoma chalconatae* n.g., n.sp (Monogenea: Polystomatidae) from *Rana chalconota* (Schlegel) of Penisular Malaysia. *Systematic Parasitology*, *49*, 223–231.

Lim, L.H.S., & Gibson, D.I. (2007). *Diplectanocotyla* Yamaguti, 1953 (Monogenea: Diplectanoidea) from *Megalops cyprinoides* (Broussonet) (Teleostei: Megalopidae) off Peninsular Malaysia. *Systematic Parasitology*, *67*, 101–117.

Lim, L.H.S. & Gibson, D.I. (2009). A new monogenean genus from an ephippid fish off Peninsular Malaysia. *Systematic Parasitology*, *73*, 13–25.

Lim, L.H.S., & Gibson, D.I. (2010). Species of *Neohaliotrema* Yamaguti, 1965 (Monogenea: Ancyrocephalidae) from the pomacentrid *Abudefduf vaigensis* (Quoy & Gaimard) off Pulau Langkawi, Malaysia, with a revised diagnosis of the genus and a key to its species. *Systematic Parasitology*, *77*, 107-129.

Lovato, C., Castellani, U., Zancanaro, C., & Giachetti, A. (2014). Automatic labelling of anatomical landmarks on 3D body scans. *Graphical Models*, *76*, 648-657.

Lyakh, A., Dmitrieva, E., Popyuk, M., Shikhat, O., & Melnik, A. (2017). A geometric morphometric approach to the analysis of the shape variability of the haptoral attachment structures of Ligophorus species (Platyhelminthes: Monogenea). *Ecologica Montenegrina*, *14*, 92-101.

Maas, A.L., Hannun, A.Y., & Ng., A.Y. (2013). Rectifier nonlinearities improve neural network acoustic models. Retrieved on July 12, 2018 from https://ai.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf.

Malekipirbazari, M., & Aksakalli, V. (2015). Risk assessment in social lending via random forests. *Expert Systems with Applications*, *42*, 4621-4631.

Martinez, A.M. & Benavente, R. (1998). The AR Face Database. Retrieved on July 1, 2018 from http://www.cat.uab.cat/Public/Publications/1998/MaB1998/CVCReport24.pdf.

Martinez, B., & Pantic, M. (2015). Facial landmarking for in-the-wild images with local inference based on global appearance. *Image and Vision Computing*, *36*, 40-50.

McCulloch, W.S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*(4), 115-133.

McEwen, B.F., & Marko, M. (1999). Three-dimensional transmission electron microscopy and its application to mitosis research. *Methods in Cell Biology*, *61*, 81-111.

Messer, K., Matas, J., Kittler, J., Luettin, J., & Maitre, G. (1999). XM2VTSDB: The Extended M2VTS Database. *Proceedings of the International Conference on Audio- and Video-Based Person Authentication*, 72-77.

Metha, C., Bhavsar, A., Oza, H., & Shah, S. (2018). *MySQL administrator's guide*. Livery Place, Birmingham: Packt Publishing Ltd.

Milborrow, S., Morkel, J., & Nicolls, F. (2010). The MUCT landmarked face database. *Proceedings of the Pattern Recognition Asscotiation of South Africa*, 32-34.

Min, J., Chen, Y.L., & Chai, J. (2009). Interactive generation of human animation with deformable motion models. *ACM Transactions on Graphics, 29* (1), 1-12.

Moeini, A., Faez, K., & Moeini, H. (2013). Deformable generic elastic models from a single 2D image for facial expression and large pose face together synthesis and recognition. *Proceedings of the Seventh International Conference on Distributed Smart Cameras (ICDSC)*, 1-6.

Mondéjar-Guerra, V., Garrido-Jurado, S., Muňoz-Salinas, Marín-Jiménez, M.J., & Medina-Carnicer, R. (2018). Robust identification of fiducial markers in challenging conditions. *Expert Systems with Applications*, *93*, 336-345.

Murakawa, J, Yoon, I, Hong, T., & Lank, E. (2006). Parts, image, and sketch based 3D modelling method. *Proceedings of the Third Eurographics conference on Sketch-Based Interfaces and Modelling*, 67-74.

Nair, V., & Hinton, G.E. (2010). Rectified linear units improve restricted Boltzmann machines. *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 807-814.

Nair, V. S., Nayak, M., Ramya, M. K., Sivadas, G., Ganesh, C., Devi, S. L., & Vedam, V. (2017). Detection of adherence of *Enterococcus faecalis* in infected dentin of extracted human teeth using confocal laser scanning microscope: An in vitro study. *Journal of Pharmacy & Bioallied Sciences*, *9*(Suppl 1), S41–S44.

Nordstrøm, M.M., Larsen, M., Sierakowski, J., & Stegmann, M. (2004). The IMM face database. *Environment Science and Policy for Sustainable Development*, *22*(10), 1-4.

Quinlan, J.R. (1999). Simplifying decision trees. *International Journal of Human-Computer Studies*, *51*, 497-510.

Petersson, H., Sinkvist, D., Wang, C., & Smedby, Ö. (2009). Web-based interactive 3D visualization as a tool for improved anatomy learning. *Anatomical Sciences Education*, *2*(2), 61-68.

Poddubnaya, L.G., Hemmingsen, W., & Gibson, D.I. (2016). Ultrastructural observations of the attachment organs of the monogenean *Rajonchocotyle emarginata* (Olsson, 1876) (Polyopisthocotylea: Hexabothriidae), a gill parasite of rays. *Parasitology Research*, *115*(6), 2285-2297.

Prentašić, P., & Lončarić. (2016). Detection of exudates in fundus photographs using deep neural networks and anatomical landmark detection fusion. *Computer Methods and Programs in Biomedicine*, *137*, 281-292.

Rahmouni, C., Vanhove, M.P.M., & Simková, A. (2017). Underexplored diversity of gill monogeneans in cichlids from Lake Tanganyika: Eight new species of *Cichlidogyrus* Paperna, 1960 (Monogenea: Dactylogyridae) from the northern basin of the lake, with remarks on the vagina and the heel of the male copulatory organ. *Parasites & Vectors*, *10*(1), 591.

Ranjan, R., Patel, V., & Chellappa, R. (2016). HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1-16.

Ranjan, R., Sankaranarayanan, S., Bansal, A., Bodla, N., Chen, J.C., Patel, V.M., … Chellappa, R. (2018). Deep learning for understanding faces: Machines may be just as good, better, than humans. *IEEE Signal Processing Magazine*, *35*(1), 66-83.

Ranjan, R., Sankaranarayanan, S., Castilo, C.D., & Chellappa, R. (2017). An all-in-one convolutional neural network for face analysis. P*roceedings of the 12th IEEE International Conference on Automatic & Gesture Recognition (FG2017)*, 17-24.

Řehulková, E., Mendlová, M., & Šimková,, A. (2013). Two new species of Cichlidogyrus (Monogenea: Dactylogyridae) parasitizing the gills of African cichlid fishes (Perciformes) from Senegal: morphometric and molecular characterization. *Parasitology Research*, *112*(4), 1399-1410.

Ren, S., Cao, X., Wei, Y., & Sun, J. (2016). *IEEE Transactions on Image Processing*, *25*(3), 1233-1245.

Rhee, T., Neumann, U., & Lewis, J.P. (2006). Human hand modeling from surface anatomy. *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 27–34.

Rodríguez-González, A., Sarabeev, V., & Balbuena, J.A. (2017). Evolutionary morphology in shape and size of haptoral anchors in 14 *Ligophorus* spp. (Monogenea: Dactylogyridae). *PLoS ONE*, *12*(5), e0178367.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386-408.

Ruthensteiner, B., & Heb, M. (2008). Embedding 3D models of biological specimens in PDF publications. *Microscopy Research and Technique*, *71*(11), 778-786.

Sagonas, C., Tzimiropouos, G., Zafeiriou, S., & Pantic, M. (2013). 300 Faces in-the-wild challenge: The first facial landmark localization challenge. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 397-403.

Samara, A., Galway, L., Bond, R., & Wang, H. (2017). Affective state detection via facial expression analysis within a human–computer interaction context. *Journal of Ambient Intelligence and Humanized Computing*, 1-10.

Samuel, A.L. (1959). Some studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, *3*(3), 210-229.

Sandnes, F.E. (2018). Sketching 3D immersed experiences rapidly by hand through 2D cross sections. In Auer, M.E. & Zutin, D. (Eds.) *Online Engineering & Internet of Things. Lecture Notes in Networks and Systems*, (pp. 1001-1013). Cham, Switzerland: Springer.

Sánchez-Lozano, E., Tzimiropoulos, G., Martinex, B., Torre, F.D.I., & Valstart, M. (2018). A functional regression approach to facial landmark tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(9), 2037-2050.

Scopigno, R., Callieri, M., Dellepiane, M., Ponchio, F., & Potenziani, M. (2017). Delivering and using 3d models on the web: Are we ready? *Virtual Archaeology Review*, *8*(17), 1-9.

Sedkaoui, S., & Monino, J.L. (2016). *Big data, open data and data development*. Hoboken, NJ: John Wiley & Sons.

Shahidi, S., Bahrampour, E., Soltanimehr, E., Zamani, A., Oshagh, M., Moattari, M., & Mehdizadeh, A. (2014). The accuracy of a designed software for automated localization of craniofacial landmarks on CBCT images. *BMC Medical Imaging*, *14*, 32.

Shinn, A.P., Gibson, D.I., & Sommerville, C. (1993). An SEM study of the haptoral sclerites of the genus *Gyrodactylus* Nordmann, 1832 (Monogenea) following extraction by digestion and sonication techniques. *Systematic Parasitology*, *25*, 135-144.

Shukla, N. (2018). *Machine Learning with TensorFlow*. Shelter Island, NY: Manning Publications Co.

Spanier, J. & Oldham, K. B. (1987). *The Hyperbolic Tangent tanh(x) and Cotangent coth(x) Functions. In An Atlas of Functions*. Washington, DC: Hemisphere.

Starck, J., & Hilton, A. (2008). Model-based human shape reconstruction from multiple views. *Computer Vision and Image Understanding*, *111*, 179-194.

Steenberge, M.V., Pariselle, A., Huyse, T., Volckaert, F.A.M, Snoeks, J., & Vanhove, M.P.M. (2015). Morphology, molecules, and monogenean parasites: An example of an integrative approach to Cichlid biodiversity. *PLoS ONE*, *10*(4), e124474.

Stonebraker, M. (2010). SQL databases v. NoSQL databases. *Communications of the ACM*, *53*(4), 10-11.

Sun, M., Kohli, P., & Shotton, J. (2012). Conditional regression forests for human pose estimation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3394-3401.

Sutton, R.S., & Barto, A.G. (1998). *Reinforcement Learning: An Introduction.* Cambridge, MA: The MIT Press.

Tan, W.B., & Lim, L.H.S. (2009). *Trianchoratus longianchoratus* sp.n. (Monogenea: Ancyrocephalidae: Heteronchocleidinae) from *Channa lucius* (Osteichthyes: Channidae) in Peninsular Malaysia. *Folia Parasitologica*, *56*(3), 180-184.

Tan, W.B., Khang, T.F., & Lim, L.H.S. (2010). Morphometric analysis of Trianchoratus Price & Berry, 1966 (Monogenea: Heteronchocleinidae) from Channa spp. (Osteichthyes: Channidae) of Penisular Malaysia. *The Raffles Bulletin of Zoology*, *58*(1), 165-172.

Tang, Y., Zou, D., Li, J., & Chen, X. (2014). Local and nonlocal guidance coupled surface deformation. *Proceedings of the SIGGRAPH Asia 2014 Technical Briefs*, 1-4.

Teo, B.G., Dhillon, S.K., & Lim, L.H.S. (2013). A deformable generic 3D model of haptoral anchor of monogenean. *PLoS ONE*, *8*(10), e77650.

Teo, B.G., Sarinder, K.K.S., & Lim, L.H.S. (2010). A novel alternative method for 3D visualization in Parasitology: the construction of a 3D model of a parasite from 2D illustrations. *Tropical Biomedicine*, *27*(2), 254-264.

Tong, Y., Liu, X., Wheeler, F.W., & Tu, P.H. (2012). Semi-supervised facial landmark annotation. *Computer Vision and Image Understanding*, *116*, 922-935.

Tonsen, M., Zhang, X., Sungano, Y., & Bulling, A. (2016). Labelled pupils in the wild: a dataset for studying pupil detection in unconstrained environments. *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, 139-142.

Valverde-Islas, L.E., Arrangoiz, E., Vega, E., Robert, L., Villanueva, R., Reynoso-Ducoing, O, … Ambrosio, J.R. (2011). Visualization and 3D reconstruction of flame cells of *Taenia solium* (Cestoda). *PLoS ONE*, *6*(3), e14754.

Vandaele R., Aceto, J., Muller, M., Péronnet, F., Debat, V., Wang, C-W., … Marée, R. (2018). Landmark detection in 2D bioimages for geometric morphometrics: A multi-resolution tree-based approach. *Scientific Reports*, *8*(1), 538.

Varian, H.R. (2014). Big data: New tricks for econometrics. *Journal of Economic Perspectives*, *28*(2), 3-28.

Wang, L., Belagiannis, V., Marr, C., Theis, F., Yang, G., & Navab, N. (2015). Anatomic-landmark detection using graphical context modelling. *Proceedings of the IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, 1304-1307.

Wang, N., Gao, X., Tao, D., Yang, H., & Li, X. (2018). Facial feature point detection: A comprehensive survey. *Neurocomputing*, *275*, 50-65.

Wang, S.F., & Lai, S.H. (2008). Estimating 3D face model and facial deformation from a single image based on expression manifold optimization. In Forsyth, D., Torr, P., & Zisserman, A. (Eds.) *Computer Vision – ECCV 2008. ECCV 2008. Lecture Notes in Computer Science*, (pp. 589-602). Springer, Berlin: Heidelberg.

Williams, D.J., & Shah, M. (1992). A fast algorithm for active contours and curvature estimation. *CVGIP: Image Understanding*, *55*(1), 14–26.

Wright, E. (2018). *Microsoft Azure IaaS solutions*. Sebastopol, CA: O'Reilly Media.

Wu, Y., Hassner, T., Kim, K., Medioni, G., & Natarajan, P. (2017). Facial landmark detection with tweaked convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(12), 3067-3074.

Xu, X.J., & Kakadiaris, I.A. (2017). Joint Head Pose Estimation and Face Alignment Framework Using Global and Local CNN Features. *Proceedings of the 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, 642-649.

Yamaguti, S. (1963). *Systema Helminthium Volume IV. Mononenea and Aspidocotylea*. London: Interscience Press.

Zaghbani, S., Boujneh, N., & Bouhlel, M.S. (2018). Age estimation using deep learning. *Computers and Electrical Engineering*, *68*, 337-347.

Zhang, J., Sun, R., Zhan, X., & Hong, C. (2015). Accurate human face pose recovery from single image through generic shape regularization. *Signal Processing*, *110*, 5-14.

Zhang, J., Gao, Y., Gao, Y., Munsell, B.C., & Shen, D. (2016). Detecting anatomical landmarks for Fast Alzheimer's Disease diagnosis. *IEEE Transactions on Medical Imaging*, *35*(12), 2524-2533.

Zhang, Z., Luo, P., Loy, C.C., & Tang, X. (2016). Learning deep representation for face alignment with auxiliary attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(5), 918-930.

Zhang, J., Wang, C., Li, C., & Qin, H. (2019). Example-based rapid generation of vegetation on terrain via CNN-based distribution learning. The Visual Computer, *35*(6), 1181-1191.

Zhao, K.W., Luximon, A., Ganesan, B., & Chan, C.K. (2015). A Composite Method for Human Foot Structural Modeling. *Procedia Manufacturing*, *3*, 3759-3766.

Zheng, S.X., Li, J., & Sun, Q.F. (2011). A novel 3D morphing approach for tooth occlusal surface reconstruction. *Computer-Aided Design*, *43*(3), 293-302.

Zhong, L., Liu, Q., Yang, P., Liu, B., Huang, J., & Metaxas, D.N. (2012). Learning active facial patches for expression analysis. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2562-2569.

Zhu, Z., Luo, P., Wang, X., & Tang, X. (2013). Deep learning identity-preserving face space. *Proceedings of the IEEE International Conference on Computer Vision*, 113-120.

Zhu, X., & Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2879-2886.

Zorman, M., Štiglic, M.M., Kokol, P., & Malčić, I. (1997). The Limitations of decision trees and automatic learning in real world medical decision making. *Journal of Medical Systems*, *21*(6), 403-415.

LIST OF PUBLICATION AND PAPERS PRESENTED

PUBLICATIONS

1. **Teo, B.G.**, Dhillon, S.K., & Lim, L.H.S. (2013). A deformable generic 3D model of haptoral anchor of monogenean. *PLoS ONE*, *8*(10), e77650.

PAPERS PRESENTED

1. **TEO, B.G.,** Dhillon, S.K., & Lim, L.H.S. (2014). *Deformation of a generic 3-D model of haptoral anchor of monogenean using direct manipulation technique*. Paper presented at the 18th Biological Science Graduate Congress, 6-8th January 2014, Kuala Lumpur, Malaysia.

2. **TEO, B.G**., Dhillon, S.K., & Lim, L.H.S. (2012). *Developing generic 3D models for 3D visualization*. Paper presented at the 17th Biological Science Graduate Congress, 8-10 Decemeber 2012, Bangkok, Thailand.