A Deep Learning Approach to Landmark Detection in Tsetse Fly Wing Images

by

Dylan Shane Geldenhuys

Thesis presented in the partial fulfilment
of the requirement for the degree of
Master of Science (Mathematics)
at the University of Stellenbosch

Supervisors: Dr. Marijn Hazelbag and Prof. John Hargrove

December 2021

PLAGIARISM DECLARATION

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: December 2021

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to the following people for institutions for all their help, guidance and support;

My supervisor Dr. Marijn Hazelbag for his invaluable guidance, support, patience and equipping me with the scientific vigour necessary to write a paper;

My co-supervisor Prof. John Hargrove for providing this data set that he spent many years creating and for his time and effort in reviewing this thesis and our fruitful discussions;

The collaborators involved in the manuscript of this project, Shane Josias for his time, council, and attention to detail. Prof. Willie Brink, Dr. Pietro Landi, Prof. Hui Cang and Jeremy Bingham for their editorial suggestions.

The South African Department of Science and Innovation-National Research Foundation (DSI-NRF), and the South African Centre for Epidemiological Modelling and Analysis (SACEMA) for financial support;

Then I would like to thank Antonia Smallberg and Casper Schutte for our fruitful discussions and their editorial suggestions;

Last but not least, I would like to thank my family for all their support throughout the years. Without them, I would not have been able to achieve my dreams.

ABSTRACT

Single-wing images were captured from 14,354 pairs of field-collected tsetse wings of species *Glossina pallidipes* and *G. m. morsitans*, and analysed together with relevant biological recordings. To answer research questions regarding these flies, we need to locate 11 anatomical landmark coordinates (x, y) on each wing. The manual location of landmarks is time-consuming, prone to error, and simply infeasible given the number of images. Automatic landmark detection has been proposed to locate these landmark coordinates.

We developed a two-tier method using deep learning architectures to classify images and make accurate landmark predictions. The first tier used a classification convolutional neural network to remove most wings that were missing landmarks. The second tier provided landmark coordinates for the remaining wings. For the second tier, we compared direct coordinate regression using a convolutional neural network and segmentation using a fully convolutional network. For the resulting landmark predictions, we evaluate shape bias using Procrustes analysis. We employ a data-centric approach paying particular attention to consistent labelling and data augmentations in training data to improve model performance.

The classification model used for the first tier achieved perfect classification on the test set. The regression and segmentation models achieved a mean pixel distance error of 5.34 (95% CI [3,7]) and 3.43 (95% CI [1.9,4.4]) respectively on 1024 × 1280 images. Segmentation had a higher computational complexity and some large outliers. Both models showed minimal shape bias.

Using this two-tier deep learning approach, we accurately filtered damaged tsetse wings with missing landmarks and provided precise landmark coordinates for the remaining wings. We chose to deploy the regression model on the complete un-annotated data since the regression model had a lower computational cost and more stable predictions than the segmentation model.

Key words:
Two tier; Deep learning; convolutional neural network; Regression; Segmentation; Classification; Landmark detection

OPSOMMING

Enkelvlerkbeelde is geneem uit 14 354 pare veldversamelde tsetse-vlieg vlerke van spesies textit Glossina pallidipes en textit G. m. morsitans, en saam met relevante biologiese metings ontleed. Om navorsingsvrae rakende hierdie vlieë te beantwoord, moet ons 11 anatomiese landmerkkoördinate (x, y) op elke vlerk vind. Aangesien die handmatige identifisering van landmerke tydrowend en vatbaar is vir foute, het ons diepleer algoritmes geleer om die koördinate van elke landmerk op te spoor.

Ons het 'n tweeledige metode ontwikkel met behulp van diepleer argitekture om beelde te klassifiseer en akkurate voorspellings vir die landmerk te maak. Eerstens het ons 'n klassifikasie-konvolusionele neurale netwerk gebruik om die meeste vlerke wat landmerke ontbreek, te verwyder. Tweedens het ons belangrike koördinate vir die oorblywende vlerke verskaf. Vir hierdie stap het ons direkte koördinaatregressie met 'n konvolusionele neurale netwerk en segmentering met 'n volledig konvolusionele netwerk vergelyk. Vir die gevolglike landmerkvoorspellings, evalueer ons vorm sydigheid met behulp van Procrustes-analise. Ons gebruik 'n data-sentriese benadering met spesiale aandag aan konsekwente etikettering en aanvulling van modelberamingsdata om modelprestasie te verbeter.

Die klassifikasiemodel wat vir die eerste stap gebruik is, het 'n perfekte klassifikasie op die toets datastel behaal. Die regressie- en segmenteringsmodelle behaal 'n gemiddelde pixelafstandfout van 5.34 (95% CI [3,7]) en 3.43 (95% CI [1.9,4.4]) onderskeidelik op 1024×1280 beelde. Segmentasie het 'n hoër berekeningskompleksiteit en 'n paar groot uitskieters. Beide modelle het minimale vorm sydigheid getoon.

Deur hierdie tweeledige benadering tot diepleer te gebruik, het ons beskadigde tsetse-vlerke akkuraat gefiltreer met ontbrekende landmerke en presiese koördinate vir die oorblywende vlerke verskaf. Ons het gekies om die regressiemodel op die volledige ongeannoteerde data te implementeer, aangesien die regressiemodel 'n laer berekeningskoste en meer stabiele voorspellings het as die segmenteringsmodel.

Sleutelwoorde: Tweeledig; diepleer; konvolusionele neurale netwerk; Regressie; Segmentasie; Klassifikasie; Landmerkopsporing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER 1

Introduction

1.1 Background

A large body of data and research has been curated on tsetse flies over the last 100 years. The main driving factor for this research lies in the necessity to understand and control African trypanosomiasis, for which the tsetse fly is a disease vector. This disease is commonly known as sleeping sickness in humans and Nagana in livestock. It is prominent in sub-Saharan parts of Africa, where livestock farming forms a significant part of the economy. In livestock, the disease induces extreme fatigue and high fever, in cattle it is known to catastrophically decrease fertility and milk production, leading to reduced reproduction rates and often death [1]. This results in significant losses in the livestock farming industry [1–3]. For this reason, there have been many studies and interventions to monitor and control tsetse populations [3–5]. The predominant method for controlling trypanosomiasis is through control of the tsetse vector. Alternatively, animals are treated with trypanocidal drugs, but these have proven insufficient in the past [3]. However, it has proven relatively easy to kill tsetse in large numbers with a significant impact on trypanosomiasis [3]. Current methods are thus geared towards eradicating tsetse through various means, which have been discussed in detail [4]. Eradicating tsetse is relatively easily achieved in isolated geographical areas such as islands. Eradication is more difficult in mainland Africa where populations are often panmictic, such that they are not entirely isolated from other populations, thus increasing the risk of re-invasion [5–7]. However, It has been noted by Ebhodaghe *et al.* [6] that "due to fragmentation of tsetse habitats by anthropogenic activities, climate differences etc., it is possible for tsetse populations to become isolated on mainlands."

The genetic similarity between tsetse populations indicates the level of mixing between them. Thus Patterson and Schofield [8], quoting Krafsur and Griffiths [9], state that "Allozyme studies, for example, have revealed high levels of genetic differentiation within populations of *G. pallidipes* and other species of the morsitans group in East Africa, suggesting that these species exist as a series of relatively isolated populations". As such, current efforts in controlling and eradicating tsetse in an area involve identifying and eradicating discrete sub-populations and populations of tsetse which are sufficiently genetically isolated from other populations, such that the risk of re-invasion after eradication is mitigated [6, 10–13].

Genetic sequence analysis is, by definition, the only exact way to gauge the genetic differentiation between populations. A phenotypic alternative to genotypic analysis is to analyse wing morphometric properties influenced by genes, which have been shown to be in close accordance with genetic variation in tsetse on multiple counts [8, 10, 12].

Accordingly, there is an increasing interest in using morphometrics of tsetse wings, which quantifies the shape of wings using the location of distinct and consistent anatomical points on the wing, referred to as landmarks. Morphometric analysis is inexpensive and easy to carry out on a small scale. Furthermore, in the wider context of biological understanding, researchers have argued that phenomics, the systematic study of phenotypes, will bring about a similar revolution to genomics [14]. Phenomics may aid in understanding or categorising important biological phenomena such as disease and evolutionary fitness where genomics has been unsuccessful [15]. The morphometric analysis for tsetse flies could be a useful surrogate to genetic analysis; moreover, phonemic and genomic research are largely synergistic, and the end product of being able to characterise both aspects of biological variation simultaneously is likely to increase the power of both approaches. [14, 15]

With the growth of geometric morphometrics in the last two decades, geneticists, evolutionary biologists, ecologists, and paleobiologists have accumulated dense data sets, often collected from thousands of specimen images [16, 17]. For tsetse flies, a large data set of digitised tsetse wings has been created. This data set is rich in morphometric information, but the process of manually annotating landmarks is inherently time-consuming and prone to error. Due to these limitations, it is generally only cost-effective to use a small sample size of specimens. Current morphometric studies on tsetse have, accordingly, only been done on a small scale [6, 11, 12]. Understanding multivariate patterns of variation requires large sample sizes [14, 18]. Therefore, the need to automate landmark detection for large data sets is an urgent and necessary development to facilitate the use morphometrics on a larger scale. Since there is already an extensive data set of digitised tsetse fly wings, the need for an automated landmark detection approach is ever more pertinent. In this context, the following section motivates the current study and sets out its aims.

1.2 Motivation and aims

This particular study is important because it marks the first attempt at addressing the issue of automatic landmark detection on a large collection of wings (≥150,000 pairs) of the tsetse flies *Glossina pallidipes* Austen and *G. m. morsitans* Westwood, which were collected over an 11-year period from Rekomitjie Research Station in the Zambezi Valley of Zimbabwe. Moreover, the vast majority of the collection of flies were females, subjected to ovarian dissection – which provided a plethora of information on the age structure of the flies processed. For about 30% of the sample, flies were also dissected to see whether they were infected with the trypanosomes. Finally, about 50,000 of the flies were also subjected to nutritional analysis. The database already contains large amounts of information regarding the time and site of sampling and various measures of the climatic conditions at and around the time of sampling. Numerous papers have been based on the data from this set [19–32]. The characteristics of the wings – both their length and their state of the fray – have been important in many of the papers cited, but their detailed morphometry has not been studied. Several other studies have, however, provided information on tsetse wing morphometrics [6, 8, 10–12, 33–36].

The idea is that the landmark records from this study will be integrated with the above database and provide a rich resource for much further research. In particular, we will eventually use the landmarks combined with other features to do several exploratory analyses. One of the directions will be to analyse the variation in wing shape with respect to seasonal and long-term climate changes. These analyses should cast light on the question of whether wing morphometry can be used to typify populations of tsetse. Another aim is to typify flies (using wing morphology) to identify mixing populations of tsetse fly. This could then be used to determine the risk of re-invasion after tsetse flies have been eradicated in a high impact area.

Photographic records have already been made of the majority of the wings. As noted above, the processing of the wings – involving the photographic process, the identification, recording and collating of a collection of landmarks, is a relatively straightforward process to complete manually when done on a small scale. However, it is not feasible to do this for 300,000 wings – hence the need to automate the process.

In this specific study, we attempt to explore current deep learning methods to develop an automatic landmark detection system for a select subset of the wings for which there is corresponding readily available digitised biological data that has been quality checked. The system should provide landmarks for tsetse wings which will subsequently be integrated with the available biological data for further morphometric research. This system will provide a proof of concept, and if successful, could be applied to all image data.

For this study, we consider a data set of 14,354 pairs of tsetse fly wing images. On each intact wing, there are 11 landmarks; these are defined by the points of vein intersections indicated in Fig 1.1. The image data comes from two volumes of the larger image data set of tsetse fly wings consisting of 17 volumes in total. Each pair of wing images has corresponding biological data recorded during dissection of these flies in the laboratory. The image data set contains wings that are missing some landmarks due to damage occurring during the capturing and processing of the flies. We refer to wing images containing all landmarks as complete wing images and refer to wings with missing landmarks as incomplete. For morphometric studies, it is important to have accurate and consistently available landmarks for each fly wing to perform geometric morphometric analysis. Apart from problem of incomplete wings, this study considers two data sets as mentioned above, namely the biological and image data set of fly wings. In some cases there are misalignments in information between these two data sets.

Accordingly, this study aims to detect and remove the incomplete wing images from the data and perform automatic landmark detection for the remaining wings. Subsequently, the landmark database created will be integrated with the available biological database and be made available for further morphometric research.

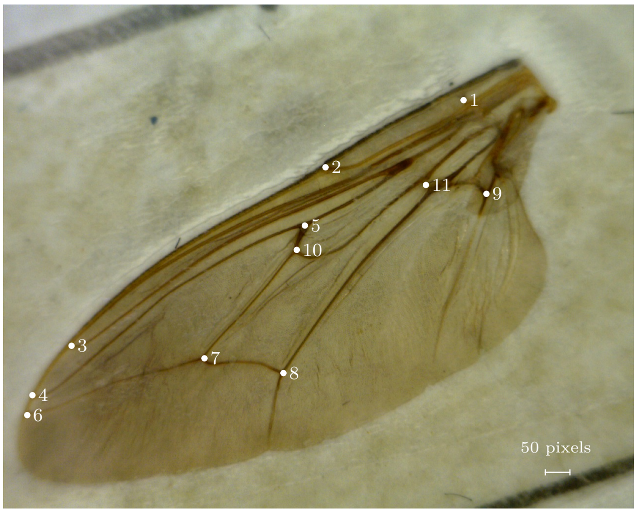


Figure 1.1. Image of a tsetse wing containing 11 landmarks indicated by white numbered points. The image also contains a scale that can be useful for placing later errors into context.

1.3 Literature review

In reviewing the literature we investigated studies that have addressed classification tasks and landmark detection on fly wings. We also pay particular attention to studies that perform landmark detection for morphometrics since it requires that landmark shape biases are avoided. For instance, in cases where landmark detection is done for robotic precision, for example, in medical operation where landmark shape is not a consideration [37]. Since there is a large body of work on landmark detection in the areas of biomedical imaging and facial recognition using deep learning, we also explore the deep learning methods within these areas. In this study we also address a data alignment issue whereby the image and biological data sets are not always linked correctly, most often occurring as misalgnments whereby an image may be shifted ahead or behind the corresponding biological data recordings. Therefore we also investigate the literature for similar problems of data alignment.

1.3.1 Related work

To our knowledge no literature addresses the problem of detecting incomplete wings for any insects or automatic landmark detection in tsetse wing images. In similar wing images, deep learning models were applied successfully in classifying fruit fly species, which are highly similar and difficult to distinguish by eye [38]. Leonardo et al. [38] compared several pre-trained convolutional neural networks for deep feature representation. The pre-trained convolutional neural networks were used to provide bottleneck features from the images, which were used as input features to train various machine learning algorithms. Although detecting incomplete wings is not of concern in this study, their methods may be a good candidate for our task since both tasks aim at wing classification. Furthermore, since these wings are highly similar, making it harder to discriminate between classes, the task complexity is most likely higher than that of classifying incomplete wings, which are fairly easy to distinguish. Therefore this approach may prove to be a powerful method for our case.

Concerning landmark detection, there have been numerous studies attempting landmark detection on other insect wings, most relevantly *Drosophila* fly wings.

Palaniswamy *et al.* [39] proposed an automatic landmark detection model for morphometrics using a data set of 856 grey-scale *Drosophila* wing images. They proposed an image analysis approach using a minimal training set of around 5 images to extract necessary information about landmarks that can be used to distinguish landmarks on other images. Their approach consisted of feature extraction followed by template matching. This approach performed well with only a small amount of manually indicated landmark data. Even so, the accuracy was highly dependent on how well the template features matched the scene features for the observed image.

Vandaele *et al.* [40] proposed a general landmark detection model for morphometrics and applied it on a *Drosophila* fly data set of 138 images of size 1440×900. Their approach used a multi-resolution tree-based approach whereby they sampled positions densely around the average position of each landmark. Haar-like features are extracted for each sample and given as inputs to a random forest classifier which predicts whether the pixel lies within a certain radius of the landmark. All the pixel coordinates that lie within this radius are averaged to decide on the final landmark position. This method uses a separate model for each landmark.

Porto and Voje [14] also proposed a general landmark detection model for morphometrics which was benchmarked on a *Drosophila* data set of 280 images of size 632×480. They employed a machine learning approach using random forests whereby object detection is used to detect the wings in the image, and a cascade regression algorithm is used to predict the landmark shape starting at an initial shape and iteratively refining the landmark shape through a cascade of regression trees. The trees use differences in pixel intensities as input variables. This method achieved better results than previous studies using semi-automated approaches [41, 42].

Contrary to this study's data set, the data set in the above-mentioned studies are comparably small. Therefore, manually chosen features are used in a machine learning or statistical approach, which can utilise small data sets effectively to train a model and make accurate landmark predictions [14, 39, 40].

Convolutional neural network architectures are often utilised on larger data sets to automatically learn the best bottleneck features due to their ability to learn complex nonlinear relationships in images, given sufficient data. In facial landmark detection and biomedical imaging, deep convolution neural network architectures have been used to obtain landmarks accurately. In facial landmark detection, the current state-of-the-art models are based on direct coordinate regression methods, where the model predicts the landmark coordinates directly, and heat-map regression methods, whereby a 2D heat map output is given for each landmark, whereby each pixel is viewed as a probability of landmark location [43]. Both these techniques have also been shown to achieve state-of-the-art results for landmark detection in biomedical image data [44–46].

In the area of facial landmark detection, facial recognition has previously been accomplished using active shape models (ASM) [47], Active Appearance Model (AAM) [48] and Constrained Local Model (CLM) [49], which all focus on deforming a mesh to place landmarks accurately. These approaches only work well in ideal scenarios with small variations, e.g. in landmark shape, expression, illumination, image blurring and occlusion. Cascade shape regressors resolved some of these problems, but the performance is difficult to improve further [50–56]. This was the approach used in the paper by Porto *et al.* [14], discussed above. More recently, deep learning networks have been employed as a more powerful method in various computer vision tasks. Various deep learning networks have been explored for landmark detection. These include Convolutional Neural Networks (CNN) [57], Auto-Encoder Network [58] and Recurrent Neural Networks (RNN) [59, 60]. Several deep learning architectures have been extensively studied and developed for landmark detection. Some of these include Fully Convolutional Neural Networks (FCN) [61], and stacked hour glass structures with residual blocks [62–64], as well as densely connected Unet architecture [65]; a type of FCN. Another crucial contribution in facial landmark detection and landmark detection in general came through considering the loss function used to train a model. Direct regression methods usually use L2 loss, which pays closer attention to large errors and struggles to reduce small errors further. Feng et al. [66] developed a new piece-wise loss function (named for its graphical shape) that switches from L1 loss to a logarithmic loss as errors become smaller. This idea was subsequently further developed for various cases [67, 68]. This ensures a steep gradient where smaller errors appear and hence more attention to reducing smaller errors close to the true landmark when using a gradient descent optimisation algorithm.

In the area of landmark detection for biomedical imaging, there has been a similar evolution using deep learning networks, and much of the breakthrough's mentioned above have been adopted in biomedical imaging [69, 70]. Most notably, landmark detection in cephalometry has similarly moved towards deep learning approaches adopting direct-coordinate regression and heat-map regression on image patches since the X-ray images are very large, often employing a further landmark refinement step using image registration or a variation of an active shape model. [45, 46, 71]

Recent papers using data-centric approaches have noted that model performance can be greatly improved by focusing on consistent and accurate data labelling for training and data augmentations [72–74]. In a data-centric approach, the model architecture is held constant while the data is iteratively improved to increase performance. This can be done by making sure labels are consistent, and that noisy samples are removed to create a clear learning task with non-ambiguous training samples - such that they clearly illustrate the concepts that need to be learned. Data augmentations are transformations that increase the data by adding slightly modified copies. It acts as a regulariser and prevents overfitting, increasing model performance and generalisability [74].

In this study, we face the problem of having two data sets that are not perfectly aligned. That is to say, wings labelled with the identical identifier in the two data sets did not refer to the same wing. In other words the data between the two data sets were not always aligned correctly. The alignment problem is discussed in numerous other studies. It has been defined as identifying and linking records that correspond to the same entity within or across several data sets [75]. It is often termed 'data linkage', 'record linkage', 'data matching', or 'entity resolution' [75–77]. A major challenge when linking records is the lack of a common identifier across the data sets. In our case, we do have a common identifier across the two data sets (image and biological), but this identifier (Volume, page, line and left or right wing) is occasionally incorrect in the image data set. When no common identifier is available, a so-called quasi-identifier (QID) is used to identify and link records about the same entity. The seminal work by Fellegi and Sunter et al. [78] on *Probabilistic data linkage* provided a sound theoretical basis [75]. They developed an optimal decision approach to classify record pairs into matches, non-matches, and potential matches based on the similarity of their QIDs [75]. In our study, most data is matched correctly and so we need to classify non-matches which we call misalignments. In many data linking problems, the data sets in question often have several QIDs such as 'Location', 'Surname' and 'Height' for medical records, but in our data we only have one viable QID; wing length. Unlike other studies described in [75], the QID in both data sets of our study were recorded differently and with different metrics, i.e. manually measured wing lengths for the biological data (in mm) and model predictions for the image data (in pixel distance), thus we do not expect a one to one relationship between wing lengths but rather some linear transformation between the two data sets. It is also worth noting that most of the misalignment's in our data occur frequently for a subset as opposed to sparsely across all the data.

1.4 Proposed approach and contributions

This thesis proposes a two-tier deep learning approach to classify tsetse fly wings with missing landmarks in the first tier followed by landmark detection in the second tier. For the first tier we compare current generic computer vision models for classification. For the second tier, we compare direct coordinate regression using a convolutional neural network and segmentation using a fully convolutional network. Fully convolutional networks have previously been used for heat-map regression with an image output size. We chose to adapt the task to output a segmentation mask in stead of a heat map. Current research has not studied segmentation for landmark detection, perhaps due to there being disproportionately more background pixels than landmark pixels. This is a well known class imbalance problem in segmentation tasks, but has been overcome in recent years due to the development of new loss functions [79–81]. As such we compare direct coordinate regression with segmentation using dice loss, which has previously been used to combat class imbalance [82–84]. The segmentation task for landmark detection is similar to that used in Vandele et al. [40] in that pixels, a given radius around the landmark, are classified, and their locations averaged to determine the final landmark location. These models were also compared with and without data augmentation. We employ a data-centric approach to develop these models, sampling training data and accurately labelling and annotating these data for classification and landmark detection. We evaluate the effect of prediction errors on subsequent morphometric analysis and apply the models to the full un-annotated data set. We further validate the classifier's performance on the un-annotated data set by comparing the expected proportion of classifications, inferred from the sample statistics, to the predicted proportion. Finally, we perform analysis for checking data alignment between the biological and image data set with predicted landmarks. We correct and remove identified errors in the data set and provide the final landmark data set as an additional contribution for subsequent morphometric studies. The workflow described is illustrated in Fig 1.2

Part of this work has been used to create a manuscript to be published in a peer-reviewed journal along with the final data set.

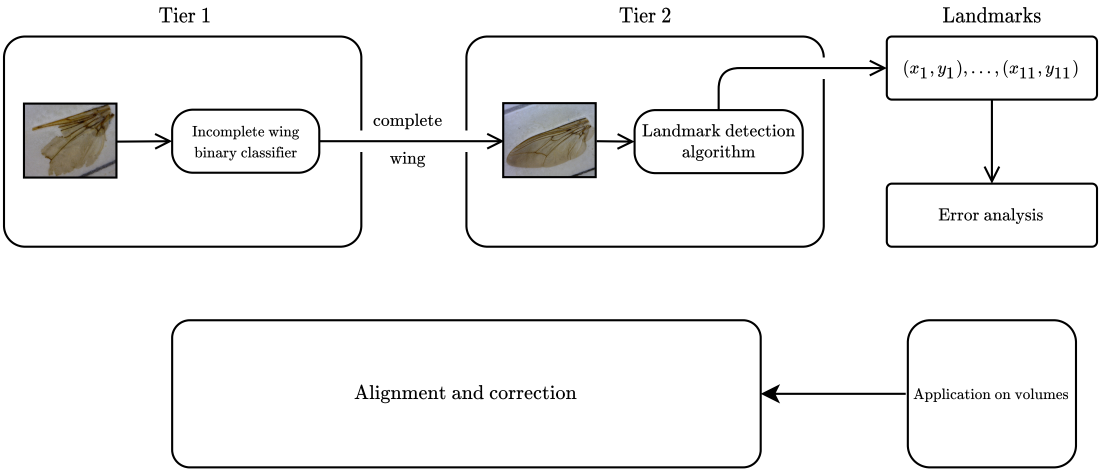

Tier 1
Incomplete wing binary classifier
complete
wing
Tier 2
Landmark detection algorithm
Landmarks
$(x_1, y_1), \ldots, (x_{11}, y_{11})$
Error analysis
Alignment and correction
Application on volumes

Figure 1.2. Flow chart of the tier 1 and 2 processes. Tier 1 decides whether a wing is complete and can be sent to tier 2 where landmarks are localised and an error analysis is performed. The final two tier system is deployed on the un-annotated data set referred (application on volumes), thereafter we detect misalignments between the image data set and biological data set, and make the necessary corrections.

CHAPTER 2

Theoretical framework

This chapter provides some of the foundational concepts needed to understand our methodological approach and why machine learning methods can achieve automatic landmark detection.

2.1 Machine learning

Machine learning is the process of fitting a model to data to approximate a mapping between the input and target space. There are many machine learning models that rely on various types of optimisation algorithms to fit the model to data. The process of fitting a model to data is termed ‘training’. Each model and optimisation algorithm has some inductive biases: that is to say, it has a set of (explicit or implicit) assumptions to generalise a finite set of observations (training data) into a general model of the domain [85]. Consider classification tasks using a decision tree model or a logistic regression model. A decision tree is always trained to overfit, fitting every training sample exactly with many axis-parallel decision boundaries in the input space. Conversely, logistic regression can only learn a single linear decision boundary across the input space.

This example also raises implicitly the concept of bias vs variance in machine learning, whereby a model with high variance is highly susceptible to changes in training data, i.e. overfitting to noisy patterns. By contrast, a model with high bias struggles to learn the necessary patterns in the data but is unaffected by small changes in training data. Hence, when developing models, it is important to consider the bias vs variance trade-off. In machine learning, it is standard practice to divide the data into three sets when developing a model. The first set is the training set, used to train the parameters of the model. The second set is used to validate model performance when training and to ensure that the model does not overfit. Overfitting occurs when the model learns the noise in the training data - resulting in an overall decrease in model performance. The third set is used if the model or optimisation algorithm has hyperparameters that may also be overfitted for the validation set. In which case, one will have a test set to evaluate the final model.

2.2 Neural Networks and Deep Learning

The most basic constituent of each deep learning model is the neuron (indicated by the circles in Fig 2.1. A Neural Network is somewhat similar to logistic regression. In essence, we can think of logistic regression as a neural network with no hidden layers and a single binary output.

Fig 2.1 shows a neural network with a single hidden layer of neurons that receive the network inputs. The weighted sum of inputs is given to the activation function of each neuron. There are many types of activation functions, the most common being the sigmoid function given in Fig 2.1. If we consider the neuron in Fig 2.1, we see that the activation function is nonlinear. This is a crucial part of a network's ability to model non-linear mappings.

A deep learning model contains many hidden layers of connected neurons with non-linear activation functions, which together make up a large parameter space capable of representing highly non-linear mappings from the input space to the target output space. This provides neural networks with the potential to learn complex nonlinear relationships given an appropriate optimisation algorithm and a sufficiently large amount of data to fit all parameters. There are various components of deep learning which we describe in the following subsections. For more details and understanding on deep neural learning please refer to the deep learning textbook by Goodfellow *et al.* [86].

The details of neural network training are described below.

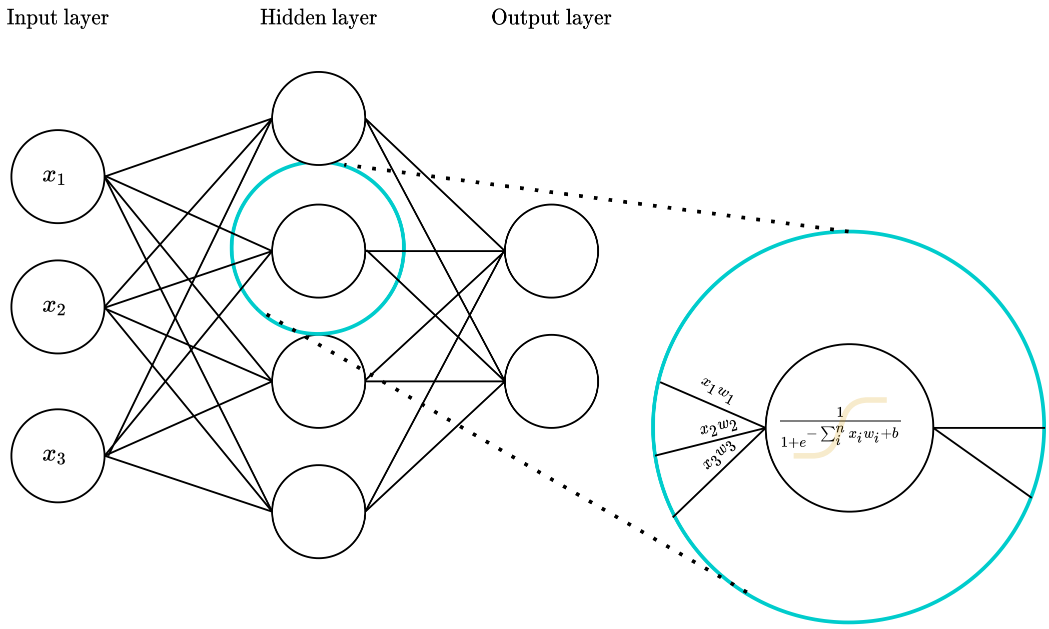


Figure 2.1. A fully connected neural network. The neurons in the input layer receive the input values directly. All other neurons receive a weighted sum of values from the previous layers, which are transformed by the activation function and provided to the next layer of neurons or as the final output.

2.2.1 Activation functions

Several activation functions have been studied. The models used in this study use two activation functions, namely the sigmoid and rectified linear unit (ReLu) [87] activation function which are defined below.

$$\text{Sigmoid} : g(z) = \frac{1}{1 + e^{-z}} \tag{2.2.1}$$

$$\text{ReLU} : g(z) = \begin{cases} 0 & \text{if } z \leq 0 \\ z & \text{if } z > 0 \end{cases} \tag{2.2.2}$$

ReLu units are very useful when using deep learning since their gradient does not saturate [86]. Unlike the ReLU function, the gradients on the sigmoid function quickly vanish (become infinitesimally small) for very large or small values. This is especially problematic when using gradient descent methods that use the chain rule for differentiation, since the product of small partial gradients quickly leads to what is called a **vanishing gradient**.

2.2.2 Layers

A deep learning model is essentially stacked layers of neurons that together form a hierarchy. Layers can be connected in various ways to provide a variety of architectures. The most standard architecture is the fully connected layer developed by Rosenblatt et al. [88]. Fig 2.1 is an example of a neural network with fully connected layers. These layers consist of neurons that receive a weighted sum of all outputs in the previous layer with an added bias term. The output of each neuron can be expressed as

$$z = g\left(b + \sum_i w_i x_i\right) \tag{2.2.3}$$

g() is the type of activation function, w_i and x_i denotes the weights and inputs, and b the added bias term.

In this study, we make use of convolutional layers developed by LeCun *et al.* [89]. Convolutional layers are particularly useful when working with images. The convolution operation uses a kernel (or filter) of weights. The kernel is shifted across the image to extract features from the image. Neurons are represented in 2D e.g $z_{m,n}$. The output of each neuron in a convolutional layer can be expressed as

$$z_{m,n} = g((f \circledast h)_{m,n}) = g\left(\sum_j \sum_k h_{j,k} f_{m-j,n-k}\right) \tag{2.2.4}$$

where f is the input image and h is the kernel. j and k refer to the number of rows and columns respectively. Kernels can be similar to filters that extract edges in images, otherwise known as Sobel operators. Early layes in a Convolutional Neural Network will extract high-level features such as edges, and extracting more technical features such as shapes and objects in deeper layers. Convolutional Neural Networks are powerful because the kernels are not predefined (as opposed to most machine learning techniques) but rather learnt. This allows deep Convolutional Neural Networks to learn optimal features from the images in the training set that can be used to make accurate predictions. Below is an example illustrating how a kernel is shifted over an image to produce the output (otherwise known as the activation map).

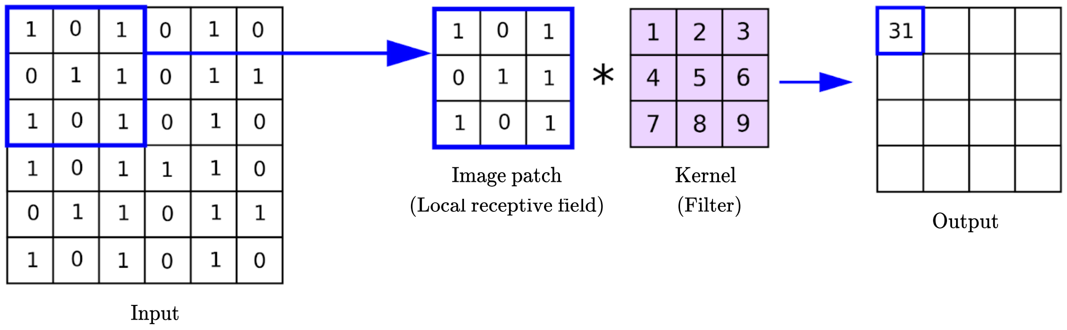


Figure 2.2. An example illustration of a convolutional operation. The kernel moves across image patches to perform convolutional operations which are provided as outputs to the activation map [90].

The hyperparameters for a convolutional layer are the **stride** and the **padding**. The stride determines the number of neurons moved over to compute the next convolutional operation, whose value is provided as the next input in the activation map. It is important to note that the stride must overlap with all neurons on the final stride along an axis: in other words, the local receptive field must always be the same size as the kernel. For instance, if the stride was 2, for the example in Fig 2.2, then on the final stride along the axis, the kernel will not be overlapping the input. To solve this we add padding to the outer layer. If we add one padding layer, then the new input will be 8×8, and then we can use a stride of 2, producing an activation map of dimension 3×3. The relationship between input size n_{in}, stride s, padding p, kernel size k and output activation map size n_{out} is formulated below.

$$z_{out} = \left(\frac{n_{in} + 2p - k}{s} \right) + 1 \tag{2.2.5}$$

2.2.3 Convolution neural networks

A Convolutional Neural Network (CNN) consists of convolutional layers with varying numbers of kernels producing multiple activation maps in the next layer as illustrated in Fig 2.3. To reduce the number of parameters and avoid overfitting, max-pooling or average pooling operations [91] are also used in CNNs. The final layers are usually flattened to provide the bottleneck features to the fully connected end of the network, which optimally maps the features to the target output space.

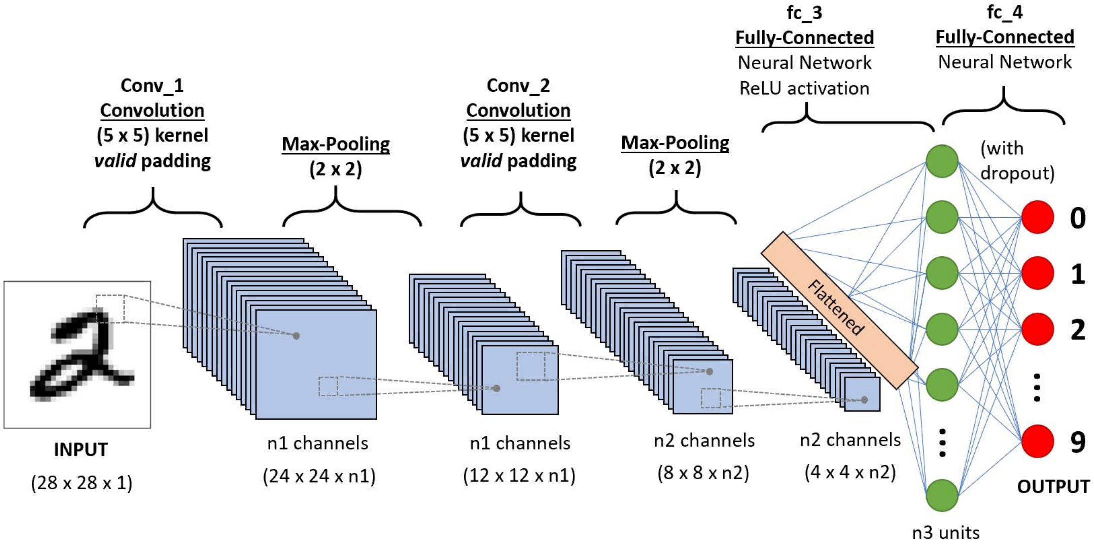


Figure 2.3. A standard CNN showing the layers and operations used to learn features. The blue blocks represent the activation maps produced from the convolutional layer, which are followed by a max-pooling operation that down-sample the activation map (reducing the dimension) [92].

In this thesis, we experiment with segmentation and regression convolutional neural networks for landmark detection. A segmentation neural network performs classification of individual pixels resulting in a segmented output image i.e in an $m \times n$ matrix of binary values, 1 usually indicates the landmark pixels and 0 non-landmark pixels. In our case, we wish to segment landmark pixels and find the location of the segmented area to extrapolate the coordinates. We can extrapolate the coordinates by finding the average location of landmark pixels i.e. the average coordinate of the segmented area. A regression convolutional neural network predicts the coordinates of the landmarks directly from the input image where all output values are real values. In the next subsections, we explore both models in more depth.

Regression convolutional neural network

A regression network provides a regression output. In this thesis, we train a regression network to provide landmark coordinate outputs. This type of network has convolutional layers, as well as fully connected layers as discussed in previous sections and illustrated in Fig 2.3. The regression network extracts information from the input image using convolutional layers. This information is provided as bottleneck features to the fully connected layers. These layers map the convolutional bottleneck features to the target space, which in this case are the landmark coordinates.

In this thesis, we use a ResNet50 architecture to perform regression [93]. ResNet50 is a state-of-the-art convolutional neural network that has been used for various classification and regression tasks and is especially stable due to its residual connections. These residual connections increase training stability resulting in improved convergence.

Deep neural networks naturally learn hierarchical features through the network. Early layers extract low-level features and deeper ones extract more high-level features. Increasing the depth of a neural network has been shown to increase the accuracy due to the added layers enriching the levels of features [93]. However, very deep neural networks are hard to train. At some stage, the added layers drastically degrade the model performance. One reason for this is believed to be due to gradient instability when using gradient-based optimisation methods. The deeper the neural network, the more gradient instability there is, causing exploding or vanishing gradients. This has largely been solved by using initialisation methods and ReLU activation functions which do not saturate. Nonetheless, for deeper networks the degradation problem still exists [93].

The construction solution for ResNet, as explained in [93], is that larger networks should be able to learn the same task as a shallow network without degradation by simply copying the shallow network with identity mappings between additional layers. However, identity mappings are not easily learnt. Accordingly, the formulation of the residual connections aims to make it easy for identify mappings to form between some layers.

The formulation of a residual layer is given in Fig 2.4

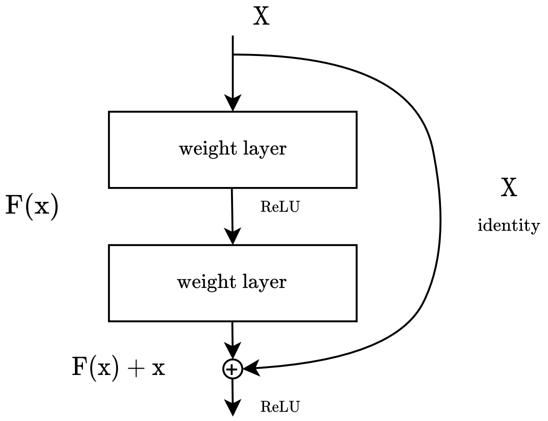


Figure 2.4. A residual connection bypassing two network layers, allowing an identity mapping to be easily approximated by reducing the residual $F(X)$ to 0. This figure was recreated from [92].

The residual connection provides an easy solution to optimally learn an identity mapping, since the weight layers can rather be pushed to zero: in other words, it decreases the residual $F(x)$ to zero. Without residual connections the network will instead have to rely on learning an identity mapping through stacked nonlinear layers.

The result of the residual layers significantly reduces the degradation problem and allows for much deeper networks to be used. This ultimately allows the network to learn deeper layers of representations which significantly increases model accuracy.

The architecture for ResNet50 is given in Appendix A, along with the full description.

Segmentation convolutional neural network

The idea of a segmentation network is to produce a mask output the same size as the image that locates and highlights target areas in the image which are objects or landmarks of interest. In practice, segmentation networks do not contain a fully connected layer: instead, they are most often **fully convolutional**, meaning they only contain convolutional layers with some pooling operations. This type of network is called a fully convolutional network (FCN), first developed by Long *et al.* [94]. The final layer is usually a soft-max or sigmoid output that produces an output image with probability values for each pixel. These probabilities are set to 0 or 1 by a threshold.

We make use of a fully convolutional network called UNet++: A Nested U-net Architecture for Medical Image Segmentation [95]. This network is a variation of the UNet architecture [96]. UNet is made up of a contraction path (encoder) and an expansion path (decoder) with skip connections (also called 'residual connections') between the two paths. The contraction path is responsible for capturing global context and low-frequency content that objects of interest are usually comprised of. The expansion path is responsible for precise location of objects [96]. Typically a CNN output is obtained from a down-sampled version of the input image created through convolutions and pooling operations, which is referred to as contraction. To produce an output the same size as the input we need to up-sample again, which is referred to as expansion. To perform expansion, a convolutional layer with a fractional stride of $1/s$ is used. This undoes the down-sampling resulting from the convolutional layer [94]. The Unet architecture first down-samples using convolutional layers and pooling - contraction. The output is received by the decoder which performs convolutions, replacing the pooling operations with large deconvolutional operations. Additionally, high-resolution activation maps from the contracting path are combined with the up-sampled output indicated by the grey arrows in Fig 2.5. A successive convolution layer can then learn to assemble a more precise output based on this information."As a consequence, the expansive path is more or less symmetric to the contracting path and yields a u-shaped architecture" [96] shown in figure Fig 2.5.

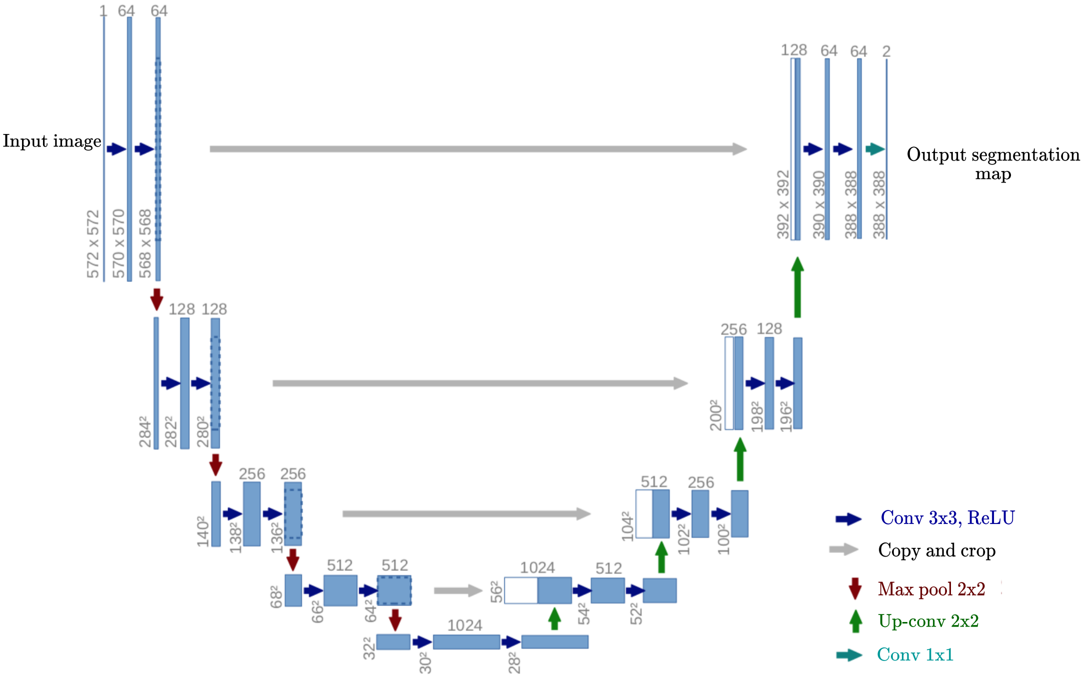


Figure 2.5. The U-Net architecture. Each blue box corresponds to multi-channel activation maps. The number of channels is given on top of the box. The size of the activation maps is provided at the lower-left edge of the box. White boxes represent copied activation maps. The arrows denote the different operations [95].

The variation of Unet used in this thesis, namely Unet++, is a deeply supervised encoder-decoder network, where the encoder and decoder sub-networks are connected through a series of nested, dense skip pathways. This creates an easier learning task for the optimiser by using redesigned skip pathways that aim at reducing the semantic gap between activation maps in the contraction and expansion paths. Zhou *et al.* [95] argue that“the optimizer would deal with an easier learning task when the activation maps from the decoder and encoder networks are semantically similar.” Fig 2.6 shows a high-level overview of the suggested architecture. Each node indicates the activation maps at a convolutional block and the black arrows indicate the down-sampling and up-sampling. UNet++ starts with an encoder sub-network (backbone) followed by a decoder sub-network. The difference between UNet++ and U-Net (the black components) is the **re-designed skip pathways** (shown in green and blue) that connect the two sub-networks, as well as the use of **deep supervision** (shown red). Both of these features are described fully in the following sections.

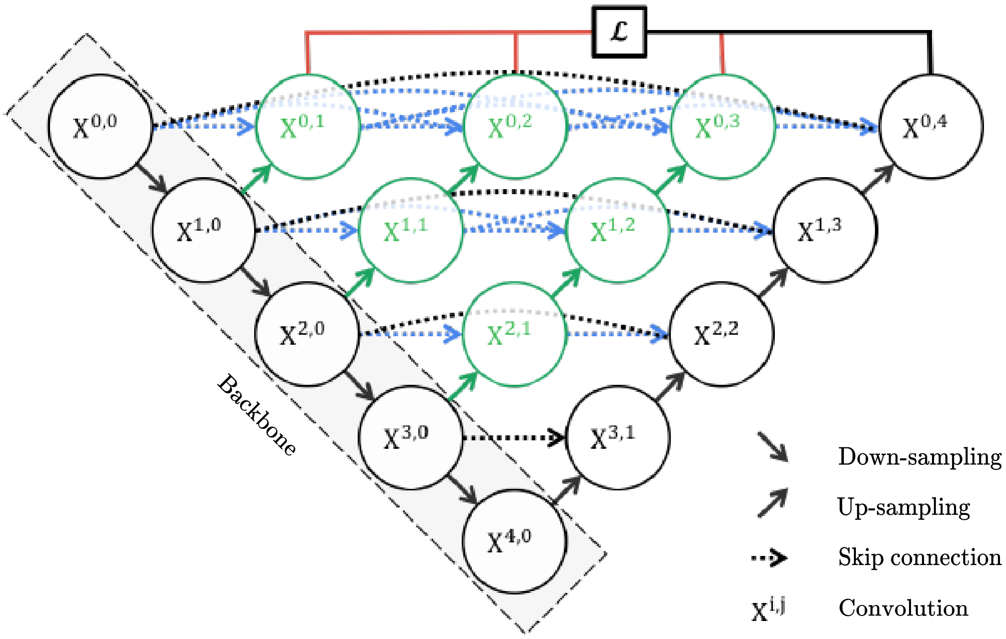


Figure 2.6. A high-level illustration of the UNet++ architecture. In the graphical illustration, black indicates the original U-Net, green and blue show dense convolution blocks on the skip pathways, and red indicates deep supervision. Red, green, and blue components distinguish UNet++ from U-Net [95].

Re-designed skip pathways

In UNet the activation maps of the encoder (output of a convolutional block) are directly received by the corresponding decoder end of the network. However, in UNet++ the activation maps from each convolutional encoder block undergo a series of convolutional layers otherwise called a dense convolutional block, whose number of convolutions depends on the pyramid layer. Each feature map (in green) is created by fusing the output activation maps of the previous convolutional layer of the same dense block with the up-sampled output of the previous convolutional layer in the lower dense block.

We formulate this as follows. Let $x^{i,j}$ denote the output of feature map $X^{i,j}$, where i indexes the down-sampled layer along the encoder and j indexes the convolutional layer of the dense block along the skip pathway. The stack of activation maps represented by $x^{i,j}$ is computed as

$$x^{i,j} = \begin{cases} H(x^{i-1,j}), & j = 0 \\ H([[x^{i,k}]_{k=0}^{j-1}, U(x^{i+1,j-1})]) & j > 0 \end{cases} \tag{2.2.6}$$

Where $H(\cdot)$ is a convolutional operation followed by an activation function, $U(\cdot)$ denotes the up-sampling layer, and [ ] denotes the concatenation layer [95].

Deep supervision

Since UNet++ generates full resolution activation maps at multiple semantic levels, the loss is estimated from 4 output activation maps, 3 indicated by the red line for skip pathways and 1 solid black line for the final output in Fig 2.6

A combination of binary cross-entropy and dice loss is used to calculate the final loss value. This loss combination is formulated below.

$$L(Y, \hat{L}) = -\frac{1}{N}\sum_{b=1}^{N}\left(\frac{1}{2} \cdot Y_b \cdot log\hat{Y_b} + \frac{2 \cdot Y_b + \hat{Y_b}}{Y_b + \hat{Y_b}}\right) \tag{2.2.7}$$

$\hat{Y_b}$ and Y_b indicate the flattened ground truths and prediction probabilities, respectively, for b^{th} image, and N indicates the number of samples.

2.2.4 Loss functions

The loss function is used to describe the error of the model. In essence, it measures the difference between the true output and the predicted output. This is then used by the optimisation algorithm to decrease the error or the loss incrementally.

In this study, we make use of three different loss functions. For classifying incomplete wings we use binary cross-entropy defined below.

$$BCE\ loss = -\frac{1}{N}\sum_{i=1}^{N} P_{pred_i}log(P_{true_i}) + (1 - P_{pred_i})log(1 - P_{true_i}) \tag{2.2.8}$$

Binary cross-entropy requires probability outputs and compares each of the predicted probabilities to actual class output which can be either 0 or 1. N refers to the number of training points and P is a probability value.

For the regression model, we use Mean Square Error defined below.

$$MSE\ loss = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.2.9}$$

The MSE loss is the most commonly used loss function for regression tasks and is especially sensitive to outliers and works well if the target data is normally distributed, since the optimal prediction will be the mean target value. $\hat{y}$ is the target value (or label) and y is the predicted value.

For the segmentation network, we use two loss functions, namely dice loss and binary cross-entropy loss. Dice loss is formulated below.

$$Dice\ loss = \frac{2\sum_i^N P_{true_i}P_{pred_i}}{\sum_i^N P_{true_i}^2 + \sum_i^N P_{pred_i}^2} \tag{2.2.10}$$

For segmentation, we have a large output with many binary outputs, one for each pixel. In our case, only a small portion of the image indicates a landmark. This means the classes of outputs are highly unbalanced. Since most pixels are non-landmarks the model could predict all pixels to be non-landmarks, achieving a high average score by only paying attention to false negatives. Dice loss is based on the Sorensen-Dice coefficient or Tversky index, which attaches similar importance to false positives and false negatives, and is more immune to the data-imbalance issue [97].

2.2.5 Optimisation algorithm

In deep learning, there are many optimisation algorithms to choose from when training a model. Most of these algorithms are based on gradient descent which use backpropagation [98] to calculate the gradient of the error function. To minimise the error by gradient descent, it is necessary to calculate the derivative with respect to all network weights; this is the sum of partial derivatives for each input-output case. Partial derivatives are calculated from a backward and forward pass. The ”forward pass” provides the error of the network from a given input. The chain rule is employed to calculate the partial derivatives in the ”back pass”, propagating derivatives backwards across the network. This process of calculating gradients is hence called backpropagation.

The loss function with all the network weights provides the loss landscape we wish to descend to reduce the error. We descend the loss landscape in a stepwise manner, estimating gradients to decide the direction of descent. The learning rate controls the step size. In this study, we use a state-of-the-art optimisation algorithm called Adam optimisation, which “computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients” [99].

The model is trained using random batches of data to estimate the current error of the network. Once the algorithm has iterated through all batches i.e. all the data, then one epoch has been completed. When we use random batches of data to estimate the error we call this type of gradient descent, stochastic gradient descent (SGD), or more specifically since we use batches we call it stochastic mini-batch gradient descent. The Adam optimiser is an extension of stochastic gradient descent.

2.2.6 Transfer learning

In this thesis, we make use of a technique called transfer learning whereby the learned features of one model is transferred or re-purposed for another related task. In other words, the knowledge or skill of a trained model is transferred to another model before training begins to improve generalisation [86]. Transfer learning works well when the features learnt on the initial task are general. We generally train the first model on a very large data set with high complexity. This increases the amount of generality in the learnt features, providing transferable features for a different task. Fig 2.7 gives an example of this process. In our case, we transfer features learnt from a very large complex data set called ImageNet with 4 million training images with 1000 classes.

Yosinski *et al.* [100] explains that“in transfer learning, we first train a base network on a base dataset and task, and then we repurpose the learned features, or transfer them, to a second target network to be trained on a target dataset and task. This process will tend to work if the features are general, meaning suitable to both base and target tasks, instead of specific to the base task.”.

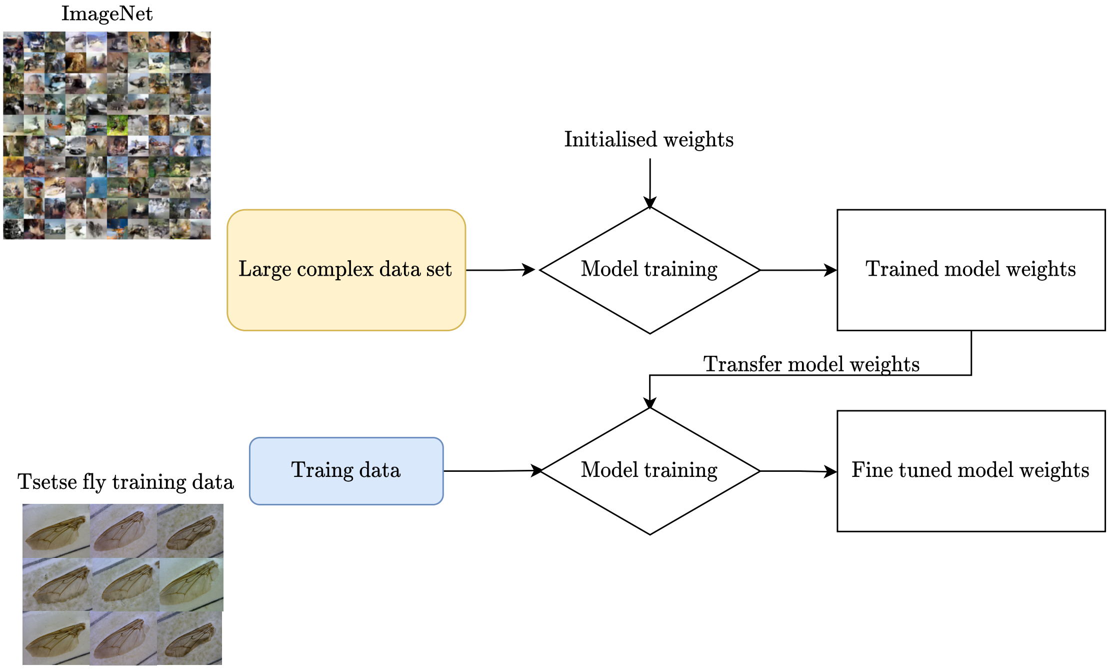


Figure 2.7. Illustration of how transfer learning is achieved in deep learning and the data sets used for transfer learning in this thesis.

Torrey and Shavlik [101] detail the benefits of transfer learning as follows:

1. **Higher start**: The initial model skill before training begins is higher than it otherwise would be.
2. **Higher slope**: The rate of improvement of skill during training of the source model is steeper than it otherwise would be.
3. **Higher asymptote**. The converged skill of the trained model is better than it otherwise would be.

There are several techniques for transfer learning: two common deep learning techniques are **feature extraction** and **finetuning** using a pretrained model.

Feature extraction involves freezing the early layers of the pretrained model so that they are not updated during training. We call this feature extraction since these layers are unchanged and use a fixed feature extractor for later layers, which are updated during training. This technique is generally used when only a small training data set is available, such as for the case in the study by Leonardo *et al.* [38].

In this thesis, we make use of the finetuning technique whereby the network is initialised with the pretrained model weights and all weights are updated during training. We call this finetuning since all the weights are further tuned from a skilled starting point.

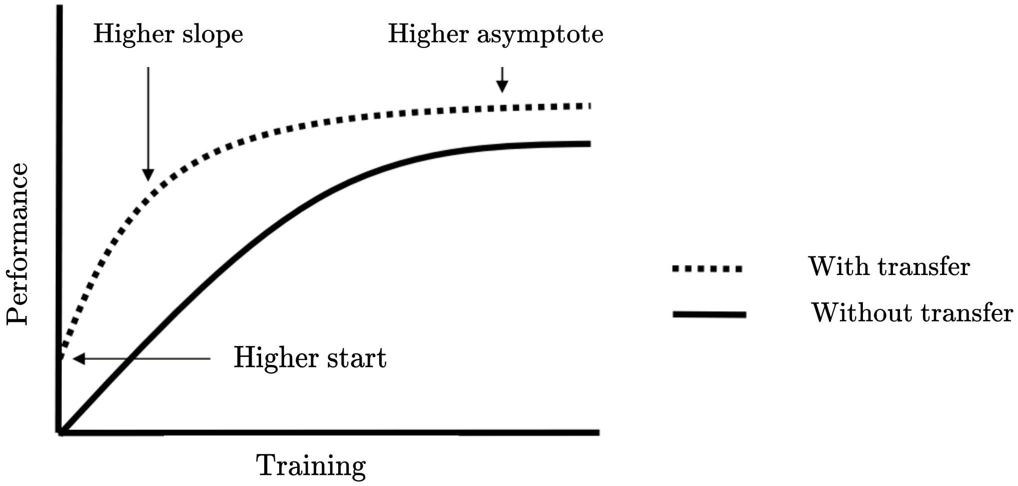


Figure 2.8. A graphical representation of the benefits of transfer learning [102].

2.3 Geometric morphometrics

In this thesis, we use geometric morphometric analysis to evaluate the predictions of our model for subsequent morphometric studies. In this section, we briefly explain the concept of Procrustes analysis as it applies to our methodological approach.

Procrustes analysis is often used to study the variation and co-variation of shapes. One way of comparing shapes is to study how they differ when the rotation, position and size of the shapes are changed to minimise the differences between them with respect to these measures, as indicated in Fig 2.9. This allows one to study the variation in shape more closely. General Procrustes analysis achieves this by centring all shapes at the origin, scales all shapes to unit size and rotates each shape around the origin until the sum of squared distances between them is minimized.

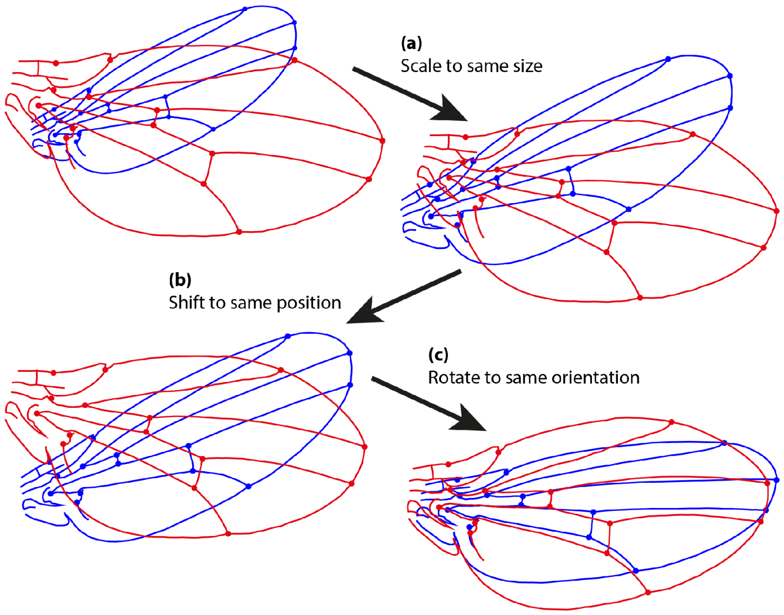


Figure 2.9. Procrustes superimposition. The figure shows the three transformation steps of an ordinary Procrustes fit for two configurations of landmarks. (a) Scaling of both configurations to the same size; (b) Transposition to the same position of the center of gravity; (c) Rotation to the orientation that provides the minimum sum of squared distances between corresponding landmarks [103].

Once two shapes are transformed so that they are the same with respect to position, rotation and size they can be compared using the Procrustes distance, which is the square root of the sum of squared differences in the positions of the landmarks in two shapes [104].

It is not uncommon for models to learn common shapes in the data with higher accuracy [67] than less common ones. This is not ideal for subsequent geometric morphometrics since the aim is to investigate changes in shape which will not be as apparent if the model is biased towards common shapes. Accordingly, in our study, we make use of Procrustes analysis to analyse this bias effect.

CHAPTER 3

Materials and methods

3.1 Data description

More than 200,000 tsetse (*Glossina pallidipes* and *G. m. morsitans*) were collected in an 11-year study carried out at Rekomitjie Research Station in the Zambezi Valley of Zimbabwe. Subsets of the flies were subjected to nutritional analysis and, for females, to ovarian dissection, to determine their age. All biological variables and descriptions can be found in Appendix B. The features used in this study for analytical purposes are given in Table 3.1.

Variable name	**Description**
vpn	Volume, page and number
wlm	Wing length measured from landmark 1 to 6
lmkl	Number of missing landmarks for left wing
lmkr	Number of missing landmarks for right wing

Table 3.1. Biological data captured in tsetse fly lab dissection.

All details relating to each collected fly were recorded on a single line of an A4 sheet of paper and the tsetse wings were fixed on this line with adhesive tape. A standard measure of wing length was determined for one wing of each pair using a binocular microscope fitted with a graduated reticule in one eyepiece. The distance between landmarks 1 and 6 (Fig 1.1) was used as this standard measure, converted to a length (wlm) in mm by allowing for magnification differences between microscopes. Each completed page was laminated between transparent plastic sheets to further protect the wings. Each page contains a maximum of 20 wing pairs. The pages were collected in 27 volumes. The wings from volumes 13 to 25 have been digitized to 1024 × 1280 resolution images using a high-resolution microscope camera. In this study, we only consider volumes 20 and 21, which refer to flies collected in 1994 and 1995. The biological data from these volumes have been digitally recorded and quality checked for morphometric studies. These volumes are also considered sufficiently large to incorporate much of the variation in all volumes and are suitable for an initial attempt at applying an automatic landmark detection system.

The image data set is kept in an identical folder structure corresponding to the physical tsetse data set. Each image is named according to the location of the physical fly wing; volume, page, line, and left- or right-wing. For example, 'V20P076L08R' refers to volume 20, page 76, line 8, right-wing. The name of each wing image links it to the biological recordings. The images were photographed from the physical data and relied on the photographer to scan the wings correctly, otherwise images would be misaligned. It is important to note that errors due to photographing the incorrect wing or incorrectly naming the page would result in misalignment between the biological and image data. Misalignment may extend to multiple wings on a page due to a single error; for example, when the photographer misses a particular wing, the upcoming sequence of photographed wings on the same page will be misaligned. Such misalignments have been noted and addressed in this study. The image data is also known to contain some missing images where there is no corresponding image for the biological data. We removed these cases when preparing the data for morphometric analysis.

To create a suitable data set for geometric morphometric analyses, we require all landmarks to be present consistently. We only considered making landmark predictions on complete wings where all 11 landmarks are visible. The types of incomplete wings fall into various categories. These include missing wings, tears, ink stains covering landmarks, and other missing pieces of the wing. Examples can be found in Appendix C. Within the class of complete wings. We also noticed a small proportion of wings with defects such as tears, ink stains, and some deformation.

The complete wings mostly vary in size, rotation, and location. Images also differ in colour contrast and brightness. Prior sample statistics were performed from a single random sample of 200 wings to understand the proportion of incomplete wings in the data. The sample statistics showed that 13% ± 4.63% of wings are incomplete. The majority of incomplete wings were missing landmark 4 or 6, making up 10% ± 4.13% of the data, and comprises on average 77% of all incomplete wings.

In our initial efforts, we attempted to remove wings using the lmkr and lmkl attributes in the biological data set, which indicate how many landmarks are missing in the right and left wing. However, visual inspection of the images accompanying these attributes revealed that the attributes were sometimes labelled incorrectly. This was likely due to either misalignment between images and biological data, or incorrect recording of missing landmarks. Therefore, we could not rely on these attributes to remove all incomplete wing images.

3.2 Workflow for landmark detection

For simplicity, we flipped the axis of all right-wing images horizontally to obtain a data set of only left wing-images. We then removed incomplete wings using a classification approach. Since most incomplete wings are missing landmarks 4 or 6, we trained a classifier to identify wings missing these landmarks. To obtain a suitable classifier, we benchmarked multiple state-of-the-art deep learning models on our data set. We then trained deep neural networks to produce landmark coordinates on complete wings. We performed image augmentations for the first tier training and experimented with and without augmentations in the second tier. For the regression model we also experiment with and without transfer learning using a model pretrained on ImageNet. To evaluate the models, we used a single cross-fold validation strategy. To examine the potential adverse effects of the landmark prediction errors on subsequent morphometric studies, we determined whether varying shapes correlated with prediction errors. To do this, we analyzed the effect of the geometric shape disparity on the average pixel distance errors. The models were then applied to all images from volumes 20 and 21. To ensure the landmark predictions and biological data were aligned correctly, we calculate the correlation between measured wing length (*wlm*) and the predicted wing length for each page. We expected a linear relationship with a high correlation between the wing lengths for well aligned pages. Accordingly, we identified pages with low R^2 values to be inspected for misalignments. The identified misalignments were corrected or removed from the data set. Finally, the measured wing length and predicted wing length were plotted for all pages and examined. The standard error was used as a measure of the agreement between the predicted and manually measured wing length. We visually examined the inliers (within prediction error interval) to inspect the predictions. Outliers were also examined for mistakes such as remaining misalignments and incorrect incomplete wing classifications. These instances were removed from the final data set.

3.3 Missing landmark classifier

3.3.1 Training data

We curated a data set consisting of a class of complete wing images containing all landmarks and a second class consisting of incomplete wing images missing landmarks 4 or 6. Since manually finding training samples is time consuming, we aided the process by using the biological data set (variable *lmkr* and *lmkl*) and then used visual inspection to filter a set of training images. We employed a data-centric approach, focusing on consistency in labelling, such that the types of wings we introduced into the training set were not ambiguous and fell into well-defined groups. We removed bad types of examples that do not clearly fall into either class. The resulting data set consists of an even class distribution of incomplete and complete wing images. Images are labelled 1 for incomplete wings and 0 for complete wings. In total, we obtained 1227 images.

3.3.2 Convolution neural network classification models

We compared three modern computer vision models with transfer learning. These models are ResNet18 [93], Inception [105], and VGG16 [106] with batch normalisation. We replaced the final fully connected layers, after the convolutional layers, with a fully connected layer of size 1. We then applied a sigmoid activation function, with an output in the range $(0, 1)$.

A 3-channel input was used for all models. Input image size of 244×244 was used for both VGG16 and ResNet18. For the Inception model, we used an input image size of 299×299 since the default kernel size of 3×3 and a stride of 2 with zero paddings require the image dimensions to be odd numbers. We fine-tuned each classifier with unfrozen weights for 30 epochs with a batch size of 50. A learning rate of 0.0001 was used with the Adam optimiser and a binary cross-entropy loss function.

Evaluation: For each classifier, we obtained 95% confidence intervals for specificity, sensitivity, precision, f1 score, and accuracy by bootstrapping the predictions on the test set for 205 samples.

3.4 Landmark detection model

3.4.1 Landmark data

To train the landmark detection models, we used a data set of 2420 complete wing images sampled from volume 20, for which landmarks were precisely annotated by a single person and subsequently reviewed by others. A custom user interface was used to digitally annotate the images and particular attention was paid to ensure landmarks were captured accurately and consistently.

Evaluation: All models were evaluated in terms of pixel distances, using mean absolute error (MAE) and root mean squared error (RMSE). The models were also compared with data augmentation turned on and off.

3.4.2 Convolutional neural network regression model

We first frame landmark detection as a regression problem, i.e. we predict the (x, y) coordinates for the landmarks directly. To that end, we considered a ResNet50 network with weights pretrained on ImageNet [107]. ResNet tends to favour a smooth loss landscape, allowing for a larger architecture while maintaining stable optimisation [93]. We altered the architecture by adding a randomly initialised convolutional layer, followed by fully connected layer with 22 outputs corresponding to each x and y coordinate for the 11 landmarks. Fig A.1 provides a high-level illustration of the model.

To train the model, we used mean square error (MSE) loss with the Adam optimisation function. The model was trained for two sessions. Each session ran for 100 epochs, initiated with a learning rate of 0.001 and reduced to 0.0001 after the first session. The model was saved at the lowest validation score in each training session.

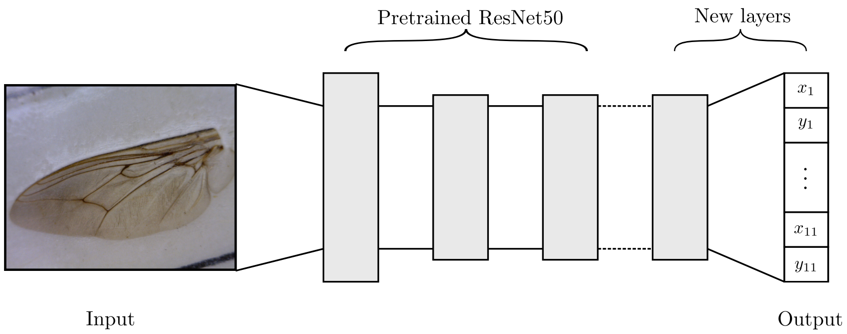


Figure 3.1. ResNet50 is modified by removing the final 2 layers and replacing them with a randomly initialised convolutional layer, followed by a fully connected layer of size 22, representing the output. The output corresponds to an x and y coordinate for each of the 11 landmarks.

3.4.3 Fully convolutional network segmentation model

As an alternative to the regression problem framing, we considered landmark detection as a semantic segmentation problem. Semantic segmentation is considered a dense prediction task, where the output is a activation map with each pixel being assigned a label. For landmark detection in tsetse fly wing images, this becomes a supervised learning problem where a label is a binary segmentation map for each landmark, with a disk centred at the (x, y) position of the landmark, as shown in Fig 3.2. The dimension of the segmentation map is the same as the dimension of the input.

For the segmentation model we chose the Unet++ [95] architecture, using the implementation as in [108].As discussed in chapter 2, Unet++ is based on the fully convolutional Unet [96] architecture that is divided into two blocks. The first is a downsampling block, responsible for capturing global context and low-frequency content that objects of interest are usually comprised of. The second is an upsampling block, responsible for precise localisation of objects [96]. Unet++ extends Unet by including skip-pathways between layers in the downsampling block and layers in the upsampling block so that the semantic gap between feature maps in those layers are reduced. Zhou et al. [95] argue that reducing the semantic gap leads to an easier optimisation problem. A high-level illustration of the architecture is shown in Fig 3.2.

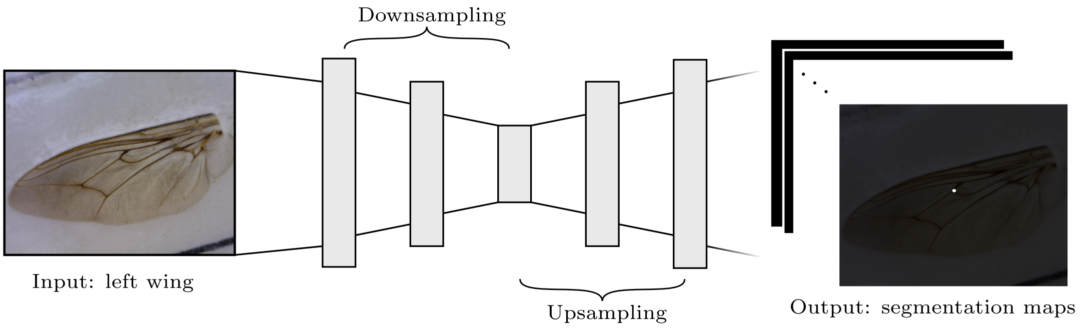


Figure 3.2. The network is composed entirely of convolutional layers. It can be divided into downsampling and symmetric upsampling blocks. The output is of dimension 11 × 224 × 224, where each output segmentation map is a binary image with a disk centred at a particular landmark.

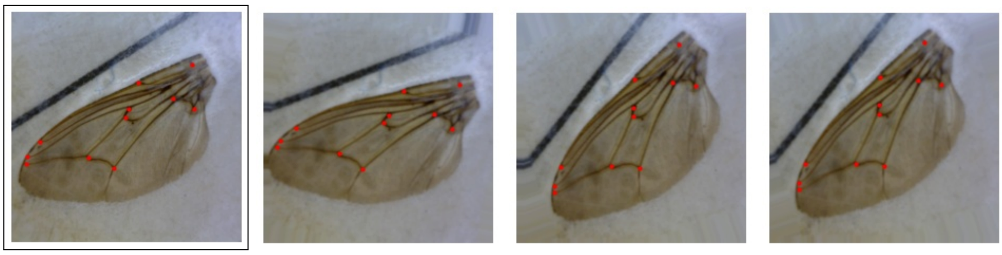
To train the model, we took the average between a binary cross entropy loss and dice loss applied to the outputs. The model was trained from scratch over 50 epochs, using the Adam optimisation function with a learning rate of 0.001. The model was saved at the lowest validation score in each training session.

To obtain landmarks from the segmentation maps, we determined the average location of all pixels greater than or equal to the seventh highest pixel value in the map. This inference process produces a landmark, even if it is absent (as does the regression network). It is possible to predict a variable number of landmarks with more a nuanced inference process, but it then becomes difficult to guarantee good generalisation.

3.5 Training implementation details

We employed transfer learning for the first tier and experimented with and without transfer learning for the regression model in the second tier [101], using networks pre-trained on ImageNet [107]. In addition, we used image augmentations during all first tier training and experimented with and without augmentations for the second tier training. Example augmentations are seen in Fig 3.3. The augmentations increased the variation in the training data, preventing over-fitting and ensuring that the training data contained images on the extreme of the spectrum for wings available in volumes 20 and 21, in terms of wing location, rotation, and size. We randomly sample a $[-5\%, 5\%]$ interval for scaling and shifting and a $[-22°, 22°]$ interval for rotation to transform each image.

We use a single cross-fold validation strategy for all model training with a 60:20:20 train, validate, and test split.



(a) Original **(b)** Aug e.g. 1 **(c)** Aug e.g. 2 **(d)** Aug e.g. 3

Figure 3.3. Example augmentations: The subplots show random augmentations produced for an example image during training. The image with the black border is the original image with no augmentations. From these figures you can see how images are rotated, shifted and scaled. The empty gaps left after augmentation are filled with border pixel of the original image

3.6 Hardware and software

The code was run on a GTX 1650 4GB GPU with a 4GHz i7 CPU and 8GB RAM. All of the code was written in Python version 3.7. The pre-trained models were taken from the Pytorch Computer Vision Library and altered for our purposes. The code can be found in a GitHub repository [109].

3.7 Effect of prediction errors on subsequent morphometric analysis

To assess the effect of prediction errors on subsequent geometric morphometrics, we evaluated the effect of the Procrustes shape disparity on the average prediction error in a wing image. In particular, we plotted the Procrustes disparity (Procrustes distance from the average shape) as the independent variable and mean pixel distance error as the dependent variable. We then fitted a regression line to determine correlation and linear relationship.

3.8 Data alignment and correction

In this study, we faced the problem of having two data sets that were not perfectly aligned. That is to say, some wings labelled with the same identifier in the two data sets did not refer to the same wing.

The approach for finding misalignments is outlined as follows. We estimated the proportion of pages containing misaligned data by generating a random sample of 100 pages and manually checking these pages for misalignments. From this we calculate an error estimate using a 95% confidence interval. We performed landmark predictions for each wing and calculated the R^2 value for each page of wings using measured wing length *wlm* vs wing length for predicted landmarks 1 and 6. We expected pages with very low R^2 values to indicate wings not aligning with the correct *wlm* measurement. We choose a R^2 threshold of 0.1, such that the percentage of pages indicated as misaligned was slightly higher than the upper bound of the confidence interval for the estimated percentage of pages misaligned. We then manually checked these pages for misaligned data. Once the pages with misaligned data were found, we considered whether we had successfully corrected or removed the misaligned data by comparing the number of pages found with the proportion estimated in the random sample.

CHAPTER 4

Results

4.1 Missing landmark classifier

After evaluating each trained model on the test set and generating a 95% confidence interval for each model, we found VGG16 with batch normalisation to have the best performance, achieving a perfect score for all metrics. ResNet18 and Inception had a trade-off between sensitivity and specificity, with ResNet18 having a perfect specificity and Inception having a perfect sensitivity. The scores for each model can be found in appendix D.

4.2 Landmark detection model

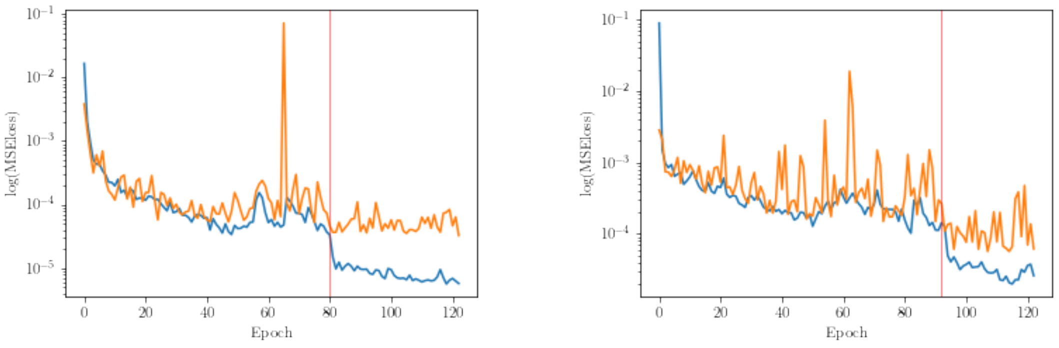
In this section we provide the results for all experiments. First we explore the effect of transfer learning used for the regression network ResNet50. Then we compare the results for pixel distance error and Procrustes disparity for the two models proposed for landmark detection. In addition we compare the effect of using data augmentation on both models.

4.2.1 Transfer learning for CNN regression

Fig 4.1 shows the log training (in blue) and log validation (in orange) loss during model training. We chose to use a semi log plot to more clearly visualise the decrease as the loss becomes smaller. Note that since we use MSE loss which is quadratic, small changes in loss create larger increases in accuracy as the loss decreases. The left and right subplots give the training and validation loss with and without transfer learning.

From the right plot we can see transfer learning provides a more skilled initialisation having a loss of 0.0028 after the first epoch, as opposed to without transfer learning which has a loss of 0.0038 after the first epoch. Transfer learning also reaches a lower asymptote faster and with more stability, reaching a validation loss of $3.1{\times}10^{-5}$ which obtains a pixel distance error of 8.3 on the test set. With no transfer learning the model reaches a loss of $4.47{\times}10^{-5}$ and obtains an average pixel distance error of 9.5 on the test set.

The red line indicates where the model was last saved in the first training session of 100 epochs and denotes the lowest validation loss score. The training with transfer learning did not benefit from the second session (starting after the red line from the last saved model). Without transfer learning the validation is reduced slightly in the second training session but training is very unstable. For both models the training loss drops noticeably lower than the validation loss for both cases. The graph also shows that at around 60 epochs the training becomes unstable for both, but the transfer learning case is able to recover.



(a) Training with transfer learning

(b) Training without transfer learning

Figure 4.1. A comparison with and without transfer learning from models that were pre-trained on ImageNet. The blue and orange lines indicate the training and validation loss after each epoch, respectively. The red vertical line denotes the position where the model was last saved in the first training session and where the second training session starts

4.2.2 CNN regression Vs FCN segmentation

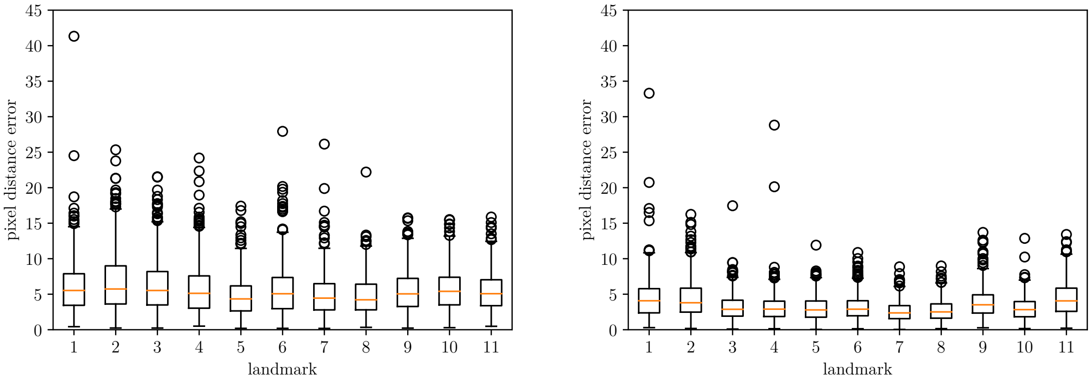
The Euclidean pixel distance predictions errors are given as a box plot in Fig 4.2. For all 11 landmarks, we obtained a mean average error (MAE) Euclidean pixel distance of 5.43 for the regression network and 3.64 for the segmentation network. For root mean average error (RMSE) we obtained 6.32 for the regression network and 6.67 for the segmentation network on the original image size (1024×1280). Since we observed skewed error distributions, we further presented the box plot of the first and third quartiles for the spread of errors. It is worth noting that there are variations among landmarks in their error distributions.

When running the models on the test set the regression network was approximately 5 times faster than the segmentation network.

Table 4.1 shows the average landmark errors for each network with and without augmentations. The segmentation network did not benefit from image augmentations but the regression network improved significantly.

	Segmentation network			Regression network		
	LB	Median	UB	LB	Median	UB
Without Augmentations	2.0	3.1	4.6	4.4	8.3	10.3
With Augmentations	1.9	3.0	4.4	3.0	4.8	7.0

Table 4.1. Effects of data augmentations on average landmark errors.



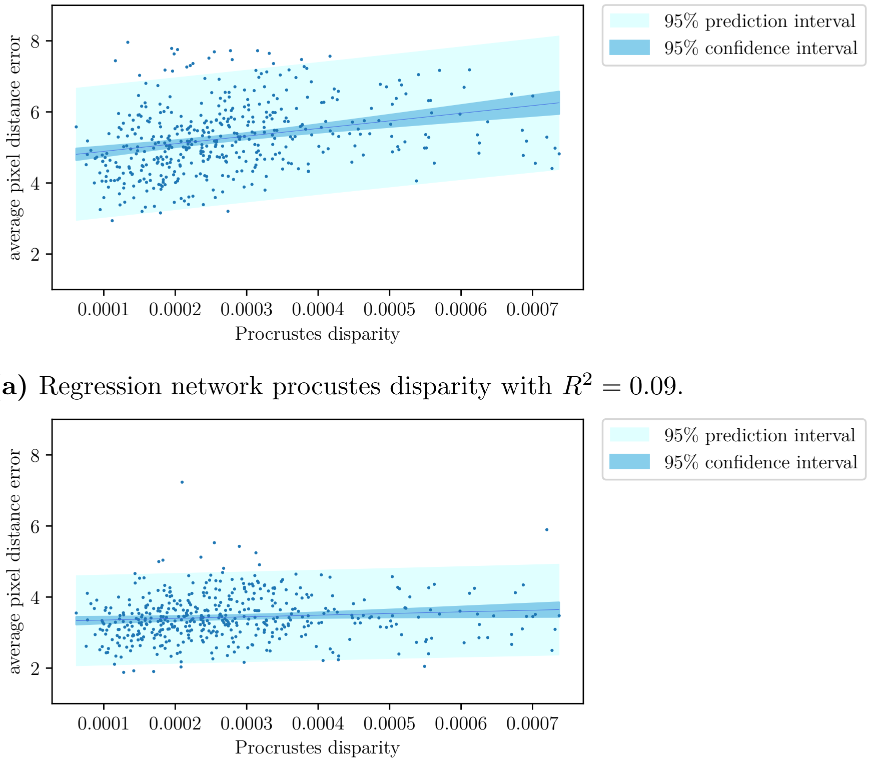
(a) The regression network has a slightly higher mean pixel distance error and higher maxima, but has fewer egregious outliers.

(b) Segmentation network. For clarity, the four outliers not displayed range from 50 to 570.

Figure 4.2. Box-and-whisker plots for the (a) regression and (b) segmentation networks.

4.3 Effect of prediction errors on subsequent morphometric analysis

Fig 4.3 shows the relationship between the mean pixel distance error, and Procrustes shape disparity from the mean shape, after removing outliers that are more than 2 standard deviations from the mean. The best fit line for regression shows a positive slope with an R^2 value of 0.09 (0.16 before removing outliers) for the regression network, and 0.01 (0.12 before removing outliers) for the segmentation network.



(a) Regression network procustes disparity with $R^2 = 0.09$.

(b) Segmentation network procrustes disparity with $R^2 = 0.01$.

Figure 4.3. Procrustes disparity from the mean landmark shape and predicted vs the mean pixel distance error.

The notable correlation can be attributed to outliers, which, if removed, brings the R^2 values down considerably for both models. Consequently, we considered the correlation weak and largely due to a few outliers. 7.2% of test data points where outliers for the regression network and 3.7% for the segmentation network. Examples of inlier predictions plotted with the ground truth are given in appendix E.

4.4 Application to volumes 20 and 21

We applied the missing landmark classifier to volumes 20 and 21. We removed all cases where the corresponding image name was missing, and the number of incomplete wings classified was 2,299 out of a total of 28,708 wings (8%). The proportion of incomplete wings removed agrees with the proportion estimated from the random sample. Visual inspection of predictions indicates that the classifier can discriminate accurately between classes, often classifying more images as incomplete as long as landmarks 4 or 6 were missing.

For the remaining complete wing images we decided to deploy the regression landmark model due to its lower computational cost and more reliable results with regard to outliers.

4.4.1 Data alignment and correction

We estimated 2% ± 2.56% (CI) of pages to be misaligned based on a random sample of 100 pages, i.e. we would expect around 15 of the 770 pages to be misaligned. Using our semi-automated procedure, we found 15 misaligned pages using the 0.1 R^2 threshold, including the two misaligned pages found in the random sample. The relevant information regarding the data cleaning can be found in appendix F.

4.4.2 Measured vs predicted wing length

Fig 4.4 shows the relationship between measured wing length (wlm) and the predicted wing length measured between landmark 1 and 6 after correcting the misaligned data. We observed a strong linear relationship between the two variables with an R^2 of 0.867. It is noted in appendix B that the wlm variable was measured for the right wing; otherwise, the hatchet cell was measured (distance between landmarks 7 and 11). We excluded all these cases in Fig 4.4. The percentage of outliers with errors larger than the standard prediction error (shown in light blue in Fig 4.4) was approximately 1.8%. The majority of outlying wings, however, received good predictions, which may indicate errors in the dissector's measurements or remaining misalignments (see an example in Fig 4.5 (i)). Other outliers represent damaged wings or missing wings that were not detected by the classifier. These outliers were expected since the classifier only considers wings missing landmark 4 or 6. The distant outliers at the bottom right of Fig 4.4 are shown in Fig 4.5 (g) and (h). These were all images with missing wings or wings that were misaligned. Fig 4.5 (g) represents a misaligned wing where the direction of the wing and image name are inconsistent. The outliers showcased in Fig 4.5 were manually removed from the final landmark data set.

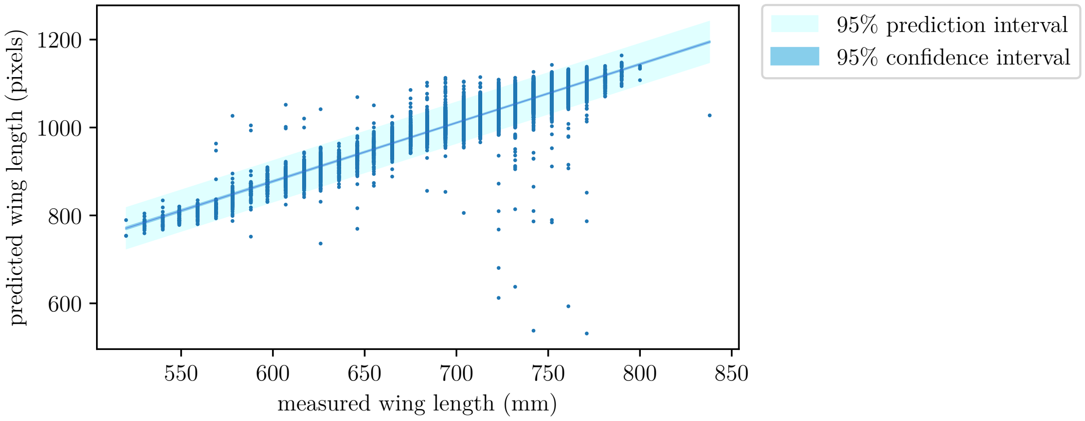
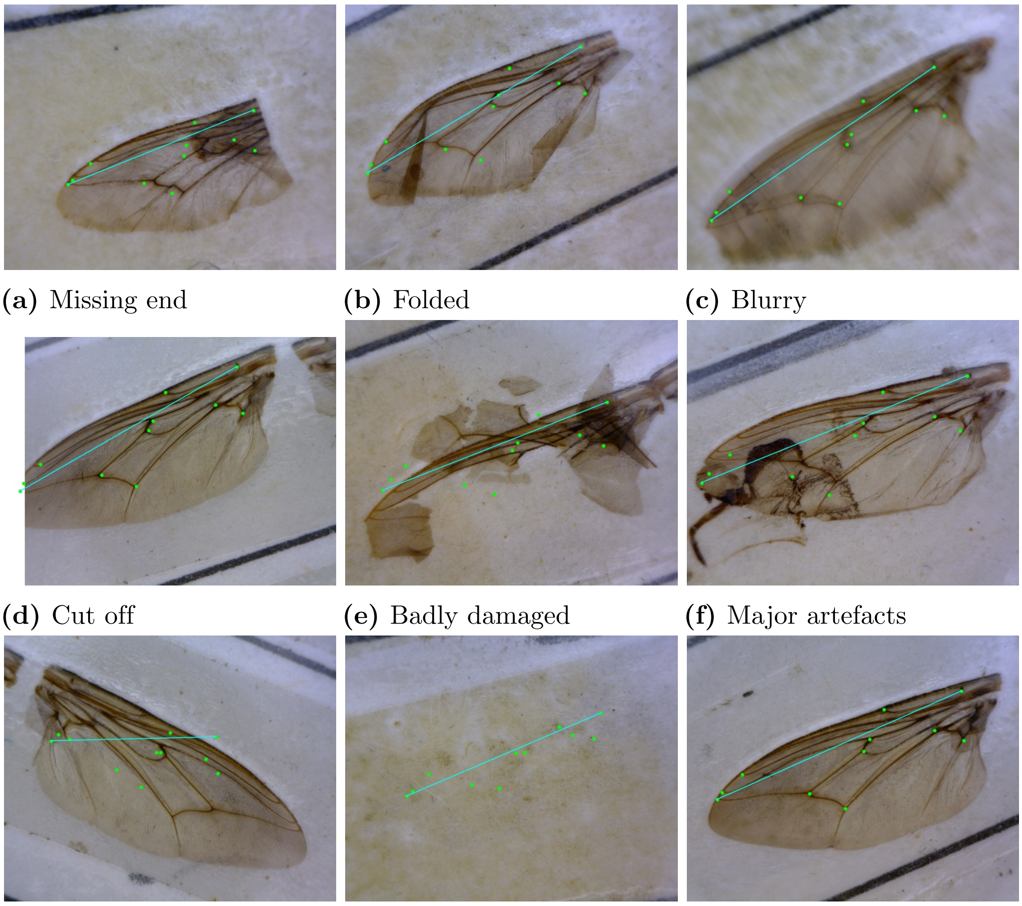


Figure 4.4. Relationship between wing length measured by the dissector vs wing length calculated from the predicted landmarks 1 and 6.



(a) Missing end

(b) Folded

(c) Blurry

(d) Cut off

(e) Badly damaged

(f) Major artefacts

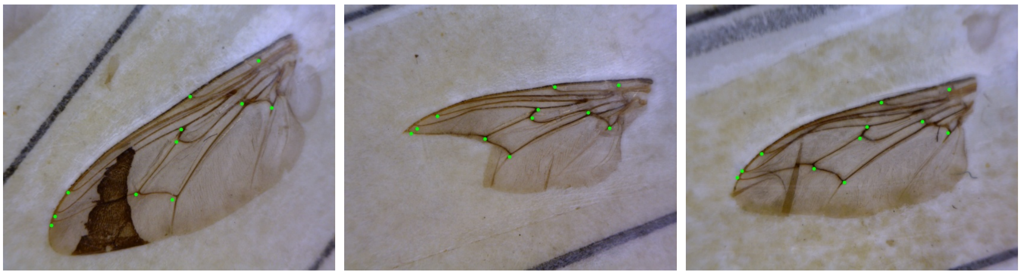
(g) Misaligned wing

(h) Missing wing

(i) Complete wing

Figure 4.5. Wing images corresponding to outliers. Dots represent predicted landmarks from the regression network and the straight line indicates wing length.

Besides examining the outliers, we also explored potential errors and the overall quality of predictions amongst the inliers. These make up the majority of wings that will be used for geometric morphometric studies. Most inlier predictions were of good quality and had accurate landmarks, with a few exceptions. Fig 4.6 shows some of the inliers with relatively large errors compared the average prediction accuracy errors, or good predictions but for deformed wings that have altered the landmark shape.



(a) Ink stain **(b)** Missing piece **(c)** Folded

Figure 4.6. Examples of Erroneous inliers. Dots represent predicted landmarks from the regression network.

CHAPTER 5

Discussion and conclusion

Using a two-tier deep learning approach, we accurately filtered out damaged tsetse wings that were missing landmarks and provided precise landmark coordinates for the remaining wings. We showed that wing shape characteristics had a minimal effect on the accuracy of landmark predictions. In addition, we addressed the misalignment problem, allowing us to make precise links between the resulting coordinates and the field-collected biological data. The result is a landmark data set with biological data that can be used for morphometric analysis.

The results indicate that our models perform slightly better than two recent machine learning approaches applied to *Drosophila* fly wings [14, 40]. Compared to our results, Vandaele et al. [40] reported a similar mean pixel distance error of approximately 6 pixels for images of a smaller size (1440 × 900). Porto et al. [14] reported a mean pixel distance error of 0.57%, normalizing the mean pixel distance by the largest wing length. Using the same metric, we obtained a smaller normalised pixel distance error of 0.47% for the deployed regression model. We noted that the *Drosophila* data sets are arguably less complex than our data. Our data often contains image artefacts and varying wing quality which may affect the complexity of the task, indicating that our model performs better than the algorithms mentioned above.

Concerning the classification task, we used a technique similar to that used by Leonardo et al. [38] for classifying fruit fly species. This study also achieved the best results using the VGG16 network compared to ResNet50, Inception and VGG19.

With regard to the data alignment problem, other approaches used the similarity of a number of qausi-identifiers (QIDs) to align data [75–78]. The QID refers to some variable that is contained in both data sets that can be used to link the data or find misalignments in the data based on their similarity. This approach would be inadequate for our case since we only have one QID (wing length) available. Moreover, dissimilarities in QID value between the data sets may be due to factors other than misalignments. Our approach was able to find misalignments successfully using a page level correlation approach using wing length as a QID.

The results of the transfer learning showed that all the benefits described in Chapter 3 were observed for our case. These were a skilled starting point, higher asymptote and faster convergence. In addition, transfer learning also gave a more stable optimisation, as seen by the close correspondence between the training and validation loss. The instability may happen because the model has optimised for the current batch of data, however this batch is not fully representative of the full data set. This results in a decrease in performance on the validation set. This is a known problem when using smaller batches [86]. Transfer learning mitigated this effect, this may be due to a more stable parameter initialisation, provided by the pre-trained network. At the start of the second training session, both models had a steeper decrease in training loss. This suggests that the model is learning patterns specific to the training data, but that does not affect the performance of the model on the validation data.

The results for the landmark model comparison indicate that, for both the landmark error and Procrustes analysis, the segmentation model performs slightly better than the regression model with regard to pixel distance error and landmark shape bias. The Procrustes analysis suggests that the segmentation model is slightly less affected by extreme changes in wing shape. After removing outliers, however, the correlation between prediction error and landmark shape is minimal for both models. The analysis also shows that the spread of errors is lower for the segmentation network, indicating higher precision. The segmentation model contains around 1.3M more parameters which amounts to a 3% increase in the number of parameters compared to the regression network. However the increased model complexity is justified by the increased task complexity created by the larger output size ($11 \times 224 \times 224$), compared to the regression output (11×2). The larger size of the segmentation model comes with extra computational cost, taking almost 5 times longer to produce predictions using a GPU while also requiring more GPU VRAM. The segmentation model has high accuracy but also produces extreme outliers, which the regression model does not. The regression model uses an MSE loss function which heavily penalises distant outliers and hence produces more consistent outputs with fewer outliers. It could also benefit from existing loss functions and regularisation, which have been shown to improve landmark detection [67]. In addition, it also showed significant improvement using data augmentations as opposed to the segmentation model in which the effects of data augmentations are insignificant.

Various workers have manually identified landmarks on tsetse wings [6, 8, 35]. To our knowledge, however, this study is the first to use automatic landmark detection on tsetse fly wing images. We employed a data-centric approach, obtaining a large sample of quality checked training images with no incorrect labels and accurate landmark annotations. In addition, we experimented with data augmentations to further increase the variety of wings regarding position, size, and rotation which are shown to improve performance for the regression model. These augmentations prevent the model from learning noisy patterns or biases in wing position, size and rotation [72]. We provide an analysis of how the prediction errors affect subsequent morphometric analysis, showing that our approach provides a model with minimal bias model concerning wing shape. This is particularly important for morphometrics since any bias the model has towards certain shapes will limit the potential for morphometric studies. This is because wing shapes at the tail of the shape distribution have less accurate landmark shape predictions, consequently affecting variation and co-variation.

A key strength of our approach is that we divided our problem into smaller tasks that allowed precise model choice with focused tasks. Since we have images with different landmarks, it may seem a suitable option to design a model to detect the available landmarks on a wing, including incomplete wings. However, this brings various complexities to the task, since it becomes more difficult to ensure that there is no shape bias between partial landmark shape predictions, i.e. how the error is affected by increasing the number of landmarks and across different landmark shapes. Creating a suitable training set may also become more difficult since we might need to include a large variety of incomplete wings, introducing sample imbalance in the training data due to varying proportions and categories of incomplete wings. In addition, from a morphometric perspective, it is standard to compare shapes with an equal number of landmarks. Our approach allowed us to easily create a training set and train a model with a fixed number of landmark outputs by first removing incomplete wings with partial landmark shapes. This avoided increased task complexity and made it directly applicable for geometric morphometrics.

There are a few limitations to the current study. First, some wings with missing landmarks or other deformities could have remained in the final data set. However, these would be small in number and unlikely to bias subsequent morphometric studies. Secondly, we cannot ensure that our R^2 threshold method achieves perfect alignment between the images and the biological data. This is because pages with only a few misalignments are less likely to be detected since they will most likely still have a high R^2 value. We are more likely to pick up misalignments due to photographing mistakes that happened at the beginning of a page, since this will mean the subsequent flies will not be in the correct order, resulting in a low R^2. However, the sample statistics for misaligned pages indicate that the majority of misalgnments were found and corrected. Furthermore, some remaining misalignments were found as outliers and removed from the final data set. Thirdly, the training images are only a subset of the images, and as such, may not include the full variety of shapes and sizes found in the full data set. However, we performed image augmentations to increase variation. Lastly, the incomplete wing classifier does not address some infrequent categories of incomplete wings, but the majority of incomplete wings are removed - leaving only a small proportion incomplete wings in the data set. Additionally, some of the remaining incomplete wings are identified as outliers for measured vs predicted wing length.

Several improvements can be made to increase overall performance of these methods and their usability on more volumes of the tsetse fly data. Future research could improve the current incomplete wing classifier by finding a method to remove all classes of incomplete wings. Alternatively, one could develop a model that predicts landmarks for both complete and incomplete wings, paying particular attention to avoid landmark shape biases in incomplete wings. Future research could improve the data augmentation to include additional affine and elastic transformations to address inaccurate predictions for divergent shapes preventing model bias towards the mean shape.

To apply the models used in this research on other volumes of the tsetse fly data, one should repeat all methodologies besides model training, i.e. calculating sample statistics and detecting misalignments to ensure the same level of data quality showcased in this study is achieved. When applying these methods to other types of data, for example, the *Drosophila* data sets, one can use transfer learning to benefit from the learnt features of this study. It is important to note that the data sets used by Vandaele et al. [40] and Porto et al. [14] will most likely not benefit from a deep learning approach because of the limited training data available. Therefore, we suggest a transfer learning approach using the trained network layers of our models.

We successfully produced landmarks on a large data set of tsetse fly wing images for subsequent morphometric studies on tsetse flies. Alongside a detailed description of the methods used, we provide the trained models and landmark data generated from this study. The trained models and landmark data can be found from the projects GitHub repository [109].

REFERENCES

[1] Swallow BM, et al. Impacts of trypanosomiasis on African agriculture. vol. 2. Citeseer; 2000.

[2] Leta S, Alemayehu G, Seyoum Z, Bezie M. Prevalence of bovine trypanosomosis in Ethiopia: a meta-analysis. Parasites & vectors. 2016;9(1):1–9.

[3] Hursey BS. The programme against African trypanosomiasis: aims, objectives and achievements. Trends in parasitology. 2001;17(1):2–3.

[4] Kuzoe FA, Schofield C, et al. Strategic review of traps and targets for tsetse and African trypanosomiasis control. TDR for research on diseases of poverty. 2005;.

[5] Hargrove J. A theoretical study of the invasion of cleared areas by tsetse flies (Diptera: Glossinidae). Bulletin of Entomological Research. 2000;90(3):201–209.

[6] Ebhodaghe F, Billah MK, Adabie-Gomez D, Yahaya A. Morphometric diagnosis of Glossina palpalis (Diptera: Glossinidae) population structure in Ghana. BMC research notes. 2017;10(1):1–6.

[7] De Meeûs T, Ravel S, Rayaisse JB, Courtin F, Solano P. Understanding local population genetics of tsetse: The case of an isolated population of Glossina palpalis gambiensis in Burkina Faso. Infection, Genetics and Evolution. 2012;12(6):1229–1234.

[8] Patterson J, Schofield C. Preliminary study of wing morphometry in relation to tsetse population genetics: research in action. South African journal of science. 2005;101(3):132–134.

[9] Krafsur E, Griffiths N. Genetic variation at structural loci in the Glossina morsitans species group. Biochemical genetics. 1997;35(1):1–11.

[10] Kaba D, Ravel S, Acapovi-Yao G, Solano P, Allou K, Bosson-Vanga H, et al. Phenetic and genetic structure of tsetse fly populations (Glossina palpalis palpalis) in southern Ivory Coast. Parasites & Vectors. 2012;5(1):1–9.

[11] Bouyer J, Ravel S, Guerrini L, Dujardin JP, Sidibé I, Vreysen MJ, et al. Population structure of Glossina palpalis gambiensis (Diptera: Glossinidae) between river basins in Burkina Faso: consequences for area-wide integrated pest management. Infection, Genetics and Evolution. 2010;10(2):321–328.

[12] Camara M, Caro-Riano H, Ravel S, Dujardin Jp, Hervouet Jp, De MeEüs T, et al. Genetic and morphometric evidence for population isolation of Glossina palpalis gambiensis (Diptera: Glossinidae) on the Loos islands, Guinea. Journal of medical entomology. 2006;43(5):853–860.

[13] Achukwi MD, Gillingwater J, Nloga AMN, Simo G, et al. Lack of evidence for sufficiently isolated populations of Glossina morsitans submorsitans on the Adamawa plateau of Cameroon following geometric morphometric analysis. Advances in Entomology. 2013;1(01):1.

[14] Porto A, Voje KL. ML-morph: A fast, accurate and general approach for automated detection and landmarking of biological structures in images. Methods in Ecology and Evolution. 2020;11(4):500–512.

[15] Houle D, Bolstad GH, van der Linde K, Hansen TF. Mutation predicts 40 million years of fly wing evolution. Nature. 2017;548(7668):447–450.

[16] Adams DC, Otárola-Castillo E. geomorph: an R package for the collection and analysis of geometric morphometric shape data. Methods in Ecology and Evolution. 2013;4(4):393–399.

[17] Adams DC, Collyer M, Kaliontzopoulou A, Sherratt E. Geomorph: Software for geometric morphometric analyses. University of New England (UNE). 2016;.

[18] Grabowski M, Porto A. How many more? Sample size determination in studies of morphological integration and evolvability. Methods in ecology and evolution. 2017;8(5):592–603.

[19] Ackley SF, Hargrove JW. A dynamic model for estimating adult female mortality from ovarian dissection data for the tsetse fly Glossina pallidipes Austen sampled in Zimbabwe. PLoS neglected tropical diseases. 2017;11(8):e0005813.

[20] English S, Cowen H, Garnett E, Hargrove JW. Maternal effects on offspring size in a natural population of the viviparous tsetse fly. Ecological Entomology. 2016;41(5):618–626.

[21] Hargrove J. Ovarian ages of tsetse flies (Diptera: Glossinidae) caught from mobile and stationary baits in the presence and absence of humans. Bulletin of Entomological Research. 1991;81(1):43–50.

[22] Hargrove J. Towards a general rule for estimating the stage of pregnancy in field-caught tsetse flies. Physiological Entomology. 1995;20(3):213–223.

[23] Hargrove J. Nutritional levels of female tsetse Glossina pallidipes from artificial refuges. Medical and Veterinary Entomology. 1999;13(2):150–164.

[24] Hargrove J. Lifetime changes in the nutritional characteristics of female tsetse Glossina pallidipes caught in odour-baited traps. Medical and Veterinary Entomology. 1999;13(2):165–176.

[25] Hargrove J. Reproductive abnormalities in tsetse flies in Zimbabwe. Entomologia Experimentalis et Applicata. 1999;92(1):89–99.

[26] Hargrove JW. Age-specific changes in sperm levels among female tsetse (Glossina spp.) with a model for the time course of insemination. Physiological entomology. 2012;37(3):278–290.

[27] Hargrove J. A model for the relationship between wing fray and chronological and ovarian ages in tsetse (Glossina spp). Medical and veterinary entomology. 2020;34(3):251–263.

[28] Hargrove JW, Ackley SF. Mortality estimates from ovarian age distributions of the tsetse fly Glossina pallidipes Austen sampled in Zimbabwe suggest the need for new analytical approaches. Bulletin of entomological research. 2015;105(3):294–304.

[29] Hargrove JW, Muzari MO. Nutritional levels of pregnant and postpartum tsetse Glossina pallidipes Austen captured in artificial warthog burrows in the Zambezi Valley of Zimbabwe. Physiological Entomology. 2015;40(2):138–148.

[30] Hargrove J, Packer M. Nutritional states of male tsetse flies (Glossina spp.)(Diptera: Glossinidae) caught in odour-baited traps and artificial refuges: models for feeding and digestion. Bulletin of Entomological Research. 1993;83(1):29–46.

[31] Hargrove JW, Muzari MO, English S. How maternal investment varies with environmental factors and the age and physiological state of wild tsetse Glossina pallidipes and Glossina morsitans morsitans. Royal Society open science. 2018;5(2):171739.

[32] Hargrove J, English S, Torr SJ, Lord J, Haines LR, Van Schalkwyk C, et al. Wing length and host location in tsetse (Glossina spp.): implications for control using stationary baits. Parasites & vectors. 2019;12(1):1–13.

[33] Getahun M, Cecchi G, Seyoum E. Population studies of Glossina pallidipes in Ethiopia: emphasis on cuticular hydrocarbons and wing morphometric analysis. Acta tropica. 2014;138:S12–S21.

[34] Bouyer J, Ravel S, Dujardin JP, De Meeüs T, Vial L, Thévenon S, et al. Population structuring of Glossina palpalis gambiensis (Diptera: Glossinidae) according to landscape fragmentation in the Mouhoun river, Burkina Faso. Journal of medical Entomology. 2007;44(5):788–795.

[35] Kaba D, Berté D, Ta B, Tellería J, Solano P, Dujardin JP. The wing venation patterns to identify single tsetse flies. Infection, Genetics and Evolution. 2017;47:132–139.

[36] Solano P, Kaba D, Ravel S, Dyer NA, Sall B, Vreysen MJ, et al. Population genetics as a tool to select tsetse control strategies: suppression or eradication of Glossina palpalis gambiensis in the Niayes of Senegal. PLoS Neglected Tropical Diseases. 2010;4(5):e692.

[37] Probst T, Maninis KK, Chhatkuli A, Ourak M, Vander Poorten E, Van Gool L. Automatic tool landmark detection for stereo vision in robot-assisted retinal surgery. IEEE Robotics and Automation Letters. 2017;3(1):612–619.

[38] Leonardo MM, Carvalho TJ, Rezende E, Zucchi R, Faria FA. Deep feature-based classifiers for fruit fly identification (diptera: Tephritidae). In: 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). IEEE; 2018. p. 41–47.

[39] Palaniswamy S, Thacker NA, Klingenberg CP. Automatic identification of landmarks in digital images. IET Computer Vision. 2010;4(4):247–260.

[40] Vandaele R, Aceto J, Muller M, Peronnet F, Debat V, Wang CW, et al. Landmark detection in 2D bioimages for geometric morphometrics: a multi-resolution tree-based approach. Scientific reports. 2018;8(1):1–13.

[41] Loh SYM, Ogawa Y, Kawana S, Tamura K, Lee HK. Semi-automated quantitative Drosophila wings measurements. BMC bioinformatics. 2017;18(1):1–14.

[42] Houle D, Mezey J, Galpern P, Carter A. Automated measurement of Drosophila wings. BMC evolutionary biology. 2003;3(1):1–13.

[43] Khabarlak K, Koriashkina L. Fast Facial Landmark Detection and Applications: A Survey. arXiv preprint arXiv:210110808. 2021;.

[44] Li J, Wang Y, Mao J, Li G, Ma R. End-to-end coordinate regression model with attention-guided mechanism for landmark localization in 3D medical images. In: International Workshop on Machine Learning in Medical Imaging. Springer; 2020. p. 624–633.

[45] Song Y, Qiao X, Iwamoto Y, Chen Yw. Automatic cephalometric landmark detection on X-ray images using a deep-learning method. Applied Sciences. 2020;10(7):2547.

[46] Zhong Z, Li J, Zhang Z, Jiao Z, Gao X. An attention-guided deep regression model for landmark detection in cephalograms. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer; 2019. p. 540–548.

[47] Cootes TF, Taylor CJ, Cooper DH, Graham J. Active shape models-their training and application. Computer vision and image understanding. 1995;61(1):38–59.

[48] Cootes TF, Edwards GJ, Taylor CJ. Active appearance models. IEEE Transactions on pattern analysis and machine intelligence. 2001;23(6):681–685.

[49] Cristinacce D, Cootes TF, et al. Feature detection and tracking with constrained local models. In: Bmvc. vol. 1. Citeseer; 2006. p. 3.

[50] Dollár P, Welinder P, Perona P. Cascaded pose regression. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE; 2010. p. 1078–1085.

[51] Xiong X, De la Torre F. Supervised descent method and its applications to face alignment. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2013. p. 532–539.

[52] Cao X, Wei Y, Wen F, Sun J. Face alignment by explicit shape regression. International journal of computer vision. 2014;107(2):177–190.

[53] Feng ZH, Huber P, Kittler J, Christmas W, Wu XJ. Random cascaded-regression copse for robust facial landmark detection. IEEE Signal Processing Letters. 2014;22(1):76–80.

[54] Wu Y, Ji Q. Constrained joint cascade regression framework for simultaneous facial action unit recognition and facial landmark detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 3400–3408.

[55] Wu Y, Gou C, Ji Q. Simultaneous facial landmark detection, pose and deformation estimation under facial occlusion. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 3471–3480.

[56] Feng ZH, Kittler J, Awais M, Huber P, Wu XJ. Face detection, bounding box aggregation and pose estimation for robust facial landmark localisation in the

wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops; 2017. p. 160–169.

[57] Sun Y, Wang X, Tang X. Deep convolutional network cascade for facial point detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2013. p. 3476–3483.

[58] Zhang J, Shan S, Kan M, Chen X. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In: European conference on computer vision. Springer; 2014. p. 1–16.

[59] Trigeorgis G, Snape P, Nicolaou MA, Antonakos E, Zafeiriou S. Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016. p. 4177–4187.

[60] Xiao S, Feng J, Xing J, Lai H, Yan S, Kassim A. Robust facial landmark detection via recurrent attentive-refinement networks. In: European conference on computer vision. Springer; 2016. p. 57–72.

[61] Liang Z, Ding S, Lin L. Unconstrained facial landmark localization with backbone-branches fully-convolutional networks. arXiv preprint arXiv:150703409. 2015;.

[62] Newell A, Yang K, Deng J. Stacked hourglass networks for human pose estimation. In: European conference on computer vision. Springer; 2016. p. 483–499.

[63] Yang J, Liu Q, Zhang K. Stacked hourglass network for robust facial landmark localisation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops; 2017. p. 79–87.

[64] Deng J, Trigeorgis G, Zhou Y, Zafeiriou S. Joint multi-view face alignment in the wild. IEEE Transactions on Image Processing. 2019;28(7):3636–3648.

[65] Tang Z, Peng X, Geng S, Wu L, Zhang S, Metaxas D. Quantized densely connected u-nets for efficient landmark localization. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 339–354.

[66] Feng ZH, Kittler J, Awais M, Huber P, Wu XJ. Wing loss for robust facial landmark localisation with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 2235–2245.

[67] Feng ZH, Kittler J, Awais M, Wu XJ. Rectified wing loss for efficient and robust facial landmark localisation with convolutional neural networks. International Journal of Computer Vision. 2020;128(8):2126–2145.

[68] Wang X, Bo L, Fuxin L. Adaptive wing loss for robust face alignment via heatmap regression. In: Proceedings of the IEEE/CVF international conference on computer vision; 2019. p. 6971–6981.

[69] Stern A, Sharan L, Romano G, Koehler S, Karck M, De Simone R, et al. Heatmap-based 2D Landmark Detection with a Varying Number of Landmarks. arXiv preprint arXiv:210102737. 2021;.

[70] Devine J, Aponte JD, Katz DC, Liu W, Vercio LDL, Forkert ND, et al. A registration and deep learning approach to automated landmark detection for geometric morphometrics. Evolutionary Biology. 2020;47(3):246–259.

[71] Arik SÖ, Ibragimov B, Xing L. Fully automated quantitative cephalometry using convolutional neural networks. Journal of Medical Imaging. 2017;4(1):014501.

[72] Zhou S, Zhang J, Jiang H, Lundh T, Ng AY. Data augmentation with Mobius transformations. Machine Learning: Science and Technology. 2021;2(2):025016.

[73] Jain S, Smit A, Ng AY, Rajpurkar P. Effect of Radiology Report Labeler Quality on Deep Learning Models for Chest X-Ray Interpretation. arXiv preprint arXiv:210400793. 2021;.

[74] Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. Journal of Big Data. 2019;6(1):1–48.

[75] Christen P. Data linkage: The big picture. Harvard Data Science Review. 2019;1(2).

[76] Christen P. Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection. Springer: Data-centric systems and applications; 2012.

[77] Herzog TN, Scheuren FJ, Winkler WE. Data quality and record linkage techniques. Springer Science & Business Media; 2007.

[78] Fellegi IP, Sunter AB. A theory for record linkage. Journal of the American Statistical Association. 1969;64(328):1183–1210.

[79] Hossain MS, Betts JM, Paplinski AP. Dual Focal Loss to address class imbalance in semantic segmentation. Neurocomputing. 2021;462:69–87.

[80] Nasalwai N, Punn NS, Sonbhadra SK, Agarwal S. Addressing the Class Imbalance Problem in Medical Image Segmentation via Accelerated Tversky Loss Function. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer; 2021. p. 390–402.

[81] Bressan PO, Junior JM, Martins JAC, Gonçalves DN, Freitas DM, Osco LP, et al. Semantic Segmentation With Labeling Uncertainty and Class Imbalance. arXiv preprint arXiv:210204566. 2021;.

[82] Sudre CH, Li W, Vercauteren T, Ourselin S, Cardoso MJ. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In: Deep learning in medical image analysis and multimodal learning for clinical decision support. Springer; 2017. p. 240–248.

[83] Li X, Sun X, Meng Y, Liang J, Wu F, Li J. Dice loss for data-imbalanced NLP tasks. arXiv preprint arXiv:191102855. 2019;.

[84] Zhao R, Qian B, Zhang X, Li Y, Wei R, Liu Y, et al. Rethinking Dice Loss for Medical Image Segmentation. In: 2020 IEEE International Conference on Data Mining (ICDM). IEEE; 2020. p. 851–860.

[85] Hüllermeier E, Fober T, Mernberger M. Inductive biases. In: Dubitzky W, Wolkenhauer O, Cho KH, Yokota H, editors. Inductive Bias. New York, NY:

Springer New York; 2013. p. 1018–1018. Available from: https://doi.org/10.1007/978-1-4419-9863-7_927.

[86] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press; 2016.

[87] Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. In: Icml; 2010. p. 807–814.

[88] Rosenblatt F. The perceptron, a perceiving and recognizing automaton Project Para. Cornell Aeronautical Laboratory; 1957.

[89] LeCun Y, Haffner P, Bottou L, Bengio Y. Object recognition with gradient-based learning. In: Shape, contour and grouping in computer vision. Springer; 1999. p. 319–345.

[90] Reynolds A. *Convolutional Neural Networks (CNNs)*; 2019. [Accessed 25th October 2021]. [Online]. Available from: https://anhreynolds.com/blogs/cnn.html.

[91] Nagi J, Ducatelle F, Di Caro GA, Cireşan D, Meier U, Giusti A, et al. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In: 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA). IEEE; 2011. p. 342–347.

[92] Saha S. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*; 2018. [Accessed 25th October 2021]. [Online]. Available from: https://towardsdatascience.com/.

[93] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–778.

[94] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 3431–3440.

[95] Zhou Z, Siddiquee MMR, Tajbakhsh N, Liang J. Unet++: A nested u-net architecture for medical image segmentation. In: Deep learning in medical image analysis and multimodal learning for clinical decision support. Springer; 2018. p. 3–11.

[96] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer; 2015. p. 234–241.

[97] Li X, Sun X, Meng Y, Liang J, Wu F, Li J. Dice loss for data-imbalanced NLP tasks. arXiv preprint arXiv:191102855. 2019;.

[98] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. nature. 1986;323(6088):533–536.

[99] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014;.

[100] Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks? arXiv preprint arXiv:14111792. 2014;.

[101] Torrey L, Shavlik J. Transfer learning. In: Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI global; 2010. p. 242–264.

[102] Brown J. A Gentle Introduction to Transfer Learning for Deep Learning; 2017. Available from: https://machinelearningmastery.com/transfer-learning-for-deep-learning/.

[103] Klingenberg CP. Analyzing fluctuating asymmetry with geometric morphometrics: concepts, methods, and applications. Symmetry. 2015;7(2):843–934.

[104] Dryden IL, Mardia KV. Statistical shape analysis: with applications in R. vol. 995. John Wiley & Sons; 2016.

[105] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 1–9.

[106] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556. 2014;.

[107] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. Ieee; 2009. p. 248–255.

[108] Bateriwala M, Bourgeat P. Enforcing temporal consistency in Deep Learning segmentation of brain MR images. arXiv preprint arXiv:190607160. 2019;.

[109] Dylan Geldenhuys, Shane Josais. *Tsetse fly landmarks for volumes 20 and 21*;. [Accessed 1rd October 2021]. [Online]. Available from: https://github.com/DylanGeldenhuys/Landmark-detection-for-tsetse-fly-wings.

[110] Ankit Sachan. *Detailed Guide to Understand and Implement ResNets*;. [Accessed 1rd October 2021]. [Online]. Available from: https://cv-tricks.com/keras/understand-implement-resnets/.

APPENDIX A

ResNet50

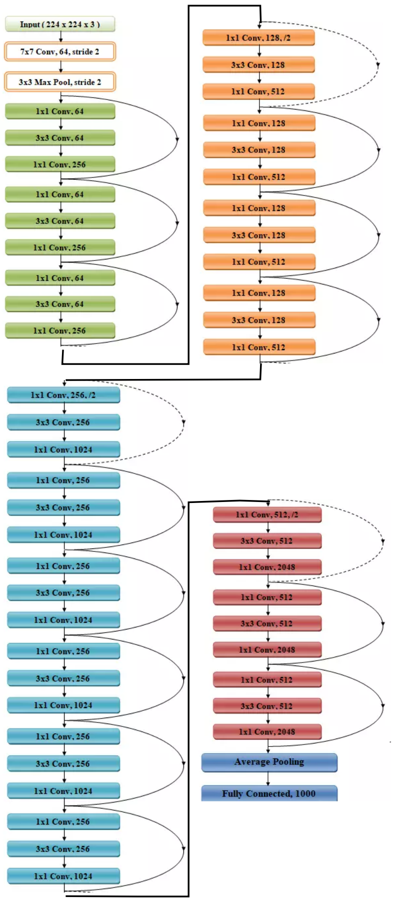


Figure A.1. The network accepts images with x,y dimensions as multiples of 32 and 3 as channel width. Consider an input size of 224×224×3. ResNet initially performs convolution and max-pooling using kernel sizes 7×7 and 3×3, respectively. Stage One has 3 Residual blocks containing three layers each. The sizes of kernels used in all three layers of the block are 64, 64 and 128, respectively. Curved arrows denote the identity connection. The dashed arrow indicates that the convolution operations in the Residual Block are performed with stride 2, reducing the size of the input to half of height and width, and outputs double the activation maps. As stages progress, the channel width is doubled, and the input size is reduced to half. Three layers are stacked between each residual connection. These layers are 1×1, 3×3, 1×1 convolutions. The 1×1 convolution layers reduce, then restore the dimensions. The 3×3 layer is a bottleneck with smaller input/output dimensions. Finally, an Average Pooling layer is used, followed by a fully connected layer having 1000 neurons (ImageNet class output). The diagram was adapted from [110].

APPENDIX B

Data collected during tsetse lab dissection

Data captured during lab dissection	
Variable name	**Description**
vpn	Volume, page and number
cd	Day of the month that fly was recorded
mc	Month of the year that fly was recorded
cy	Year that fly was recorded
md	Method used to capture fly
g	Genus of the fly [Gp = G. pallidipes or Gmm = G. m. morsitans]
s	Sex [1= Male 2 = Female]
c	Ovarian category, which measures age [1 to 7]
wlm	wing length, measured from the right wing when available
f	Wing fray category, also measures age [1 – 6]
lmkl	Number of landmarks missing on LEFT wing [1 – 11]
lmkr	Number of landmarks missing on RIGHT wing [1 – 11]
hc	Index. 1 if hatchet cell used; 0 if distance landmarks 1 to 6 used

Table B.1. The variable names given in this table were recorded during the dissection of each respective fly. Note that these variables are only some of the variables available for each fly which are given as part of the morphometric data set published with this research.

APPENDIX C

Examples of bad wings

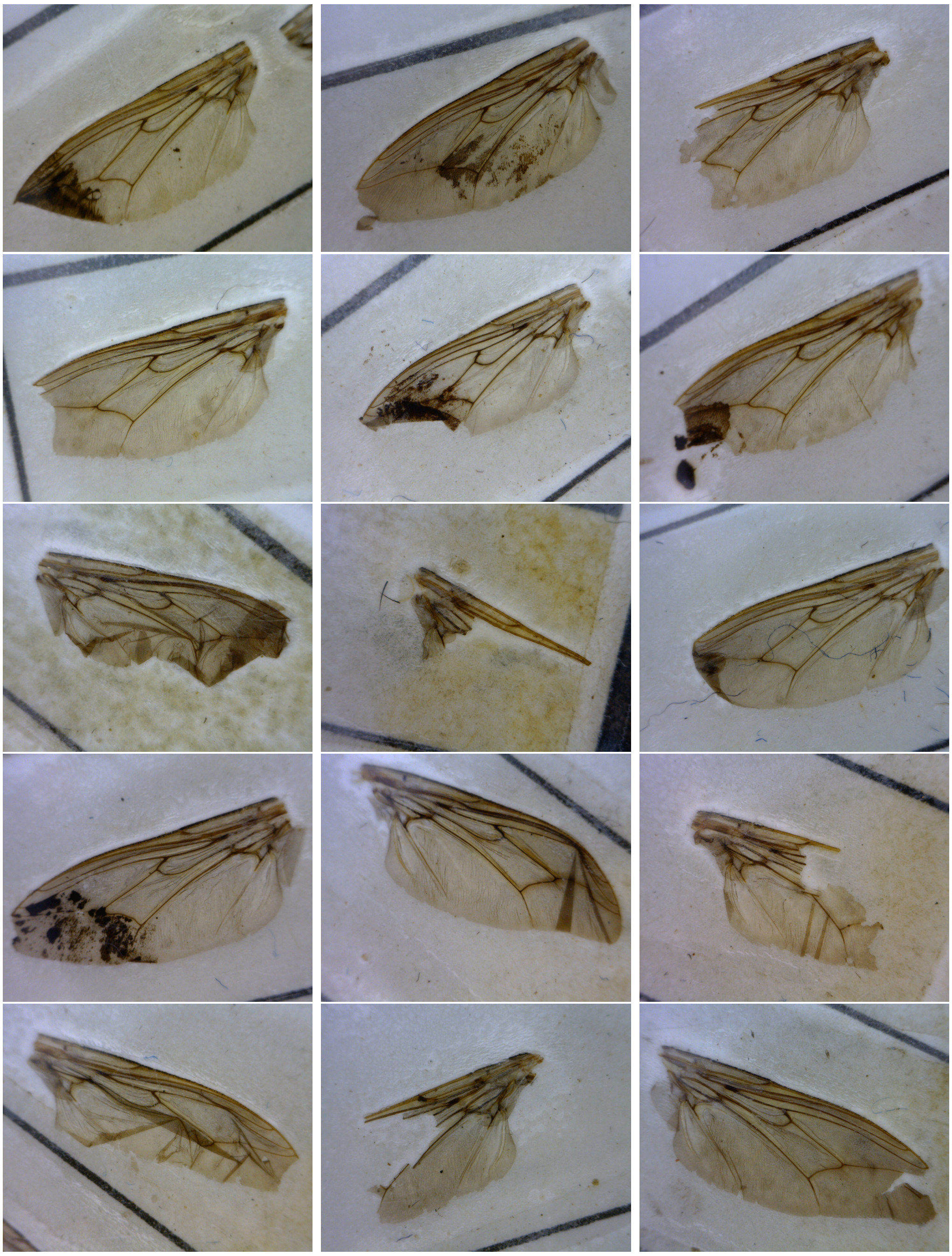


Figure C.1. Examples of the types of bad wings that appear in the data set.

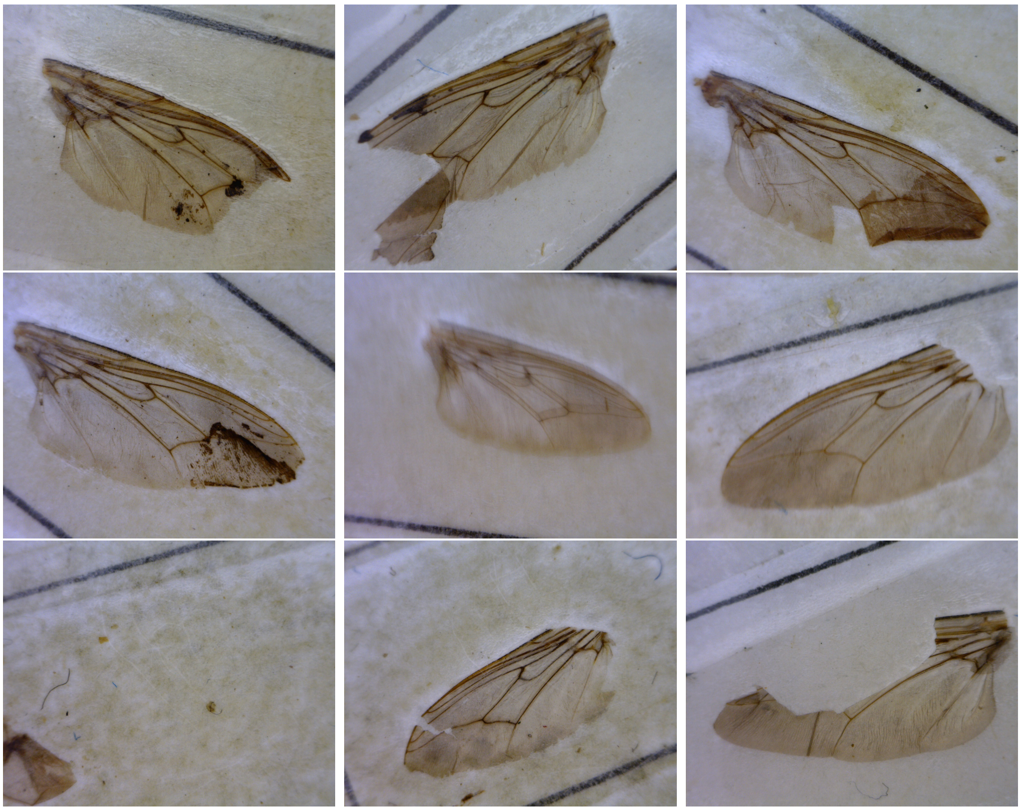


Figure C.2. Examples of the types of bad wings that appear in the data set.

APPENDIX D

Classification benchmark scores

Tables 1,2 and 3 provide the score and confidence intervals for each metric score. Tables 4 and 5 show that ResNet18 and Inception have a trade off between specificity and sensitivity with ResNet18 having a better specificity and Inception a better sensitivity. The best performing was VGG16, attaining 100% on all metrics.

Table D.1. VGG16 scores

95% confidence interval	specificity	sensitivity	precision	f1 score	accuracy
lower bound	1.000	1.000	1.000	1.000	1.000
value	1.000	1.000	1.000	1.000	1.000
upper bound	1.000	1.000	1.000	1.000	1.000

Table D.2. ResNet18 scores

95% confidence interval	specificity	sensitivity	precision	f1 score	accuracy
lower bound	1.000	0.972	1.000	0.986	0.985
value	1.000	0.993	1.000	0.991	0.990
upper bound	1.000	1.00	1.000	1.000	1.000

Table D.3. Inception V3 scores

95% confidence interval	specificity	sensitivity	precision	f1 score	accuracy
lower bound	0.957	1.000	0.961	0.980	0.980
value	0.985	1.000	0.980	0.990	0.990
upper bound	1.000	1.000	1.000	1.000	1.000

APPENDIX E

Landmark predictions on Vol. 20-21

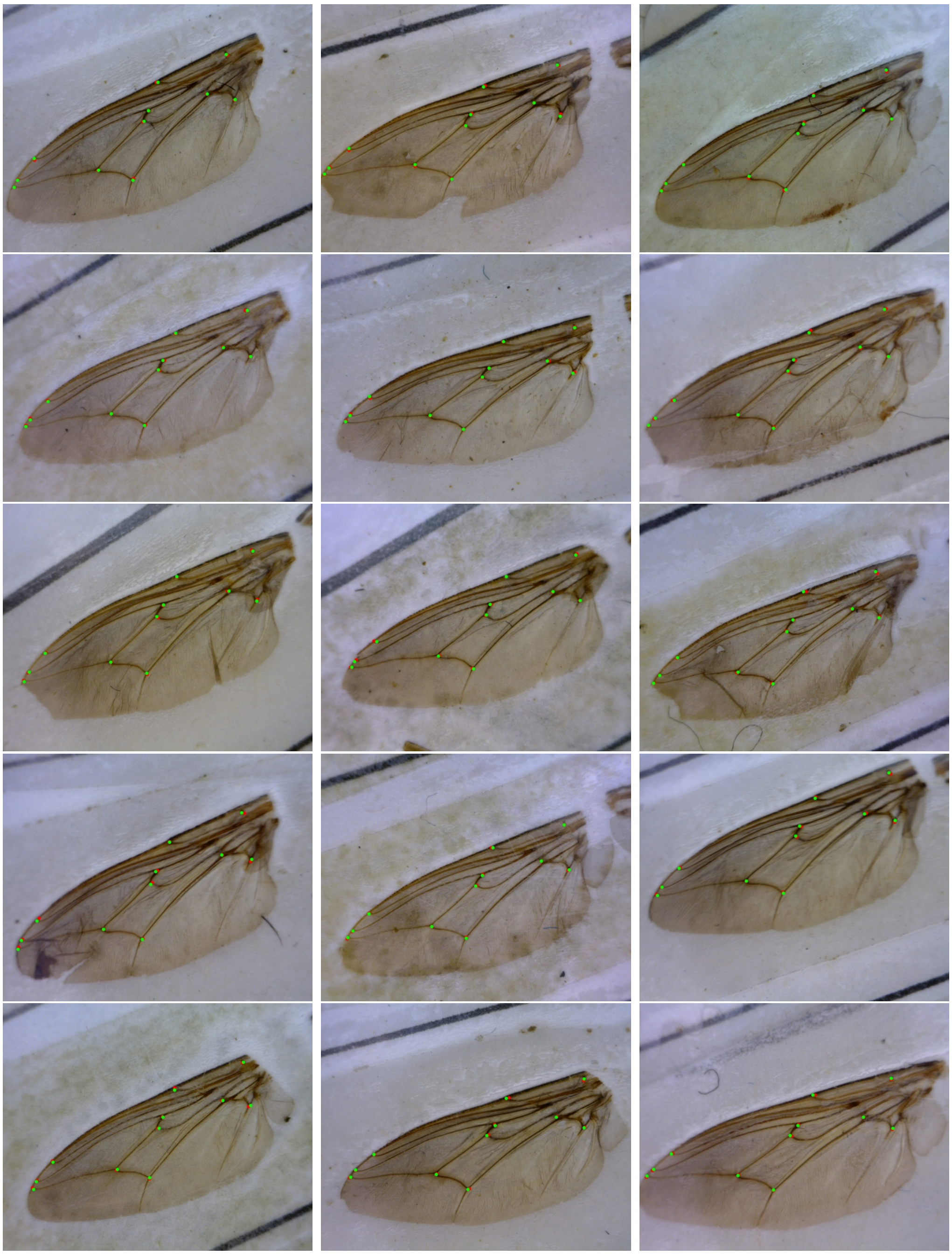


Figure E.1. Example predictions within the 95% CI for Procrustes disparity and mean pixel distance error. Green dots indicate ground truth and red dots indicate predictions.

APPENDIX F

Data information

Sample statistics for misaligned pages

sample size	100
population size	770
confidence interval	95%
pages found (%)	2
margin of error (%)	2.56

Table F.1. A table of sample statistics for pages containing mismatching data

Table 6 gives the results of how many pages where found in both volumes and either corrected or removed.

Number of pages containing miss matching data	15
Number of pages corrected	12
Number of pages removed	3

Table F.2. A table detailing the amount of mismatching pages
found and whether they were corrected or removed from the data set.

Alterations to volume 20 and 21

Pages with $R^2 < 0.1$:

- 3 pages removed - 36, 48, 321
- 12 pages corrected - 11, 34,35,42,75, 159, 187, 201, 226, 232, 357, 358.