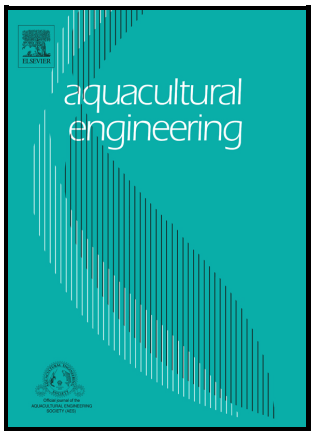
Prawn Morphometrics and Weight Estimation from Images using Deep Learning for Landmark Localization

Alzayat Saleh, Md Mehedi Hasan, Herman W. Raadsma, Mehar S. Khatkar, Dean R. Jerry, Mostafa Rahimi Azghadi



PII: S0144-8609(24)00002-5

DOI: https://doi.org/10.1016/j.aquaeng.2024.102391

Reference: AQUE102391

To appear in: *Aquacultural Engineering*

Received date: 17 July 2023
Revised date: 4 January 2024
Accepted date: 11 January 2024

Please cite this article as: Alzayat Saleh, Md Mehedi Hasan, Herman W. Raadsma, Mehar S. Khatkar, Dean R. Jerry and Mostafa Rahimi Azghadi, Prawn Morphometrics and Weight Estimation from Images using Deep Learning for Landmark Localization, *Aquacultural Engineering,* (2024) doi:https://doi.org/10.1016/j.aquaeng.2024.102391

Prawn Morphometrics and Weight Estimation from Images using Deep Learning for Landmark Localization

Alzayat Saleh*, Md Mehedi Hasan†, Herman W Raadsma†, Mehar S Khatkar†, Dean R Jerry*, and Mostafa Rahimi Azghadi *‡

*College of Science and Engineering, James Cook University, Townsville, QLD, Australia

†Sydney School of Veterinary Science, Faculty of Science, The University of Sydney, Camden, NSW, Australia

‡Corresponding author: mostafa.rahimiazghadi@jcu.edu.au

Abstract

Accurate morphometric analyses and weight estimation are useful in aquaculture for optimizing feeding, predicting harvest yields, identifying desirable traits for selective breeding, grading processes, and monitoring the health status of production animals. However, the collection of phenotypic data through traditional manual approaches at industrial scales and in real-time is time-consuming, labour-intensive, and prone to errors. Digital imaging of individuals and subsequent training of prediction models using Deep Learning (DL) has the potential to rapidly and accurately acquire phenotypic data from aquaculture species. In this study, we applied a novel DL approach to automate morphometric analysis and weight estimation using the black tiger prawn (Penaeus monodon) as a model crustacean. The DL approach comprises two main components: a feature extraction module that efficiently combines low-level and high-level features using the Kronecker product operation; followed by a landmark localization module that then uses these features to predict the coordinates of key morphological points (landmarks) on the prawn body. Once these landmarks were extracted, weight was estimated using a weight regression module based on the

extracted landmarks using a fully connected network. For morphometric analyses, we utilized the detected landmarks to derive five important prawn traits. Principal Component Analysis (PCA) was also used to identify landmark-derived distances, which were found to be highly correlated with shape features such as body length, and width. We evaluated our approach on a large dataset of 8164 images of the Black tiger prawn (Penaeus monodon) collected from Australian farms. Our experimental results demonstrate that the novel DL approach outperforms existing DL methods in terms of accuracy, robustness, and efficiency.

Index Terms

Weight Estimation, Computer Vision, Morphometric Analyses, Convolutional Neural Networks, Machine Learning, Deep Learning, Aquaculture.

1 INTRODUCTION

The farming of marine prawns (shrimp) is one of the most important and largest aquaculture production sectors globally [1]. As in any animal production sector, the acquisition of industrial-scale data on weight and other commercially relevant phenotypic traits is important, as this data can improve yields and economic efficiency through informing pond management, feeding, grading and selective breeding processes [2–7]. However, the current collection of this data is manual using traditional weight and morphometric analyses that are often invasive, time-consuming, labour-intensive, and prone to human error. Therefore, there is a need for the development of automated, fast, and accurate methods for weight estimation and associated morphometric analyses.

Computer vision and image analysis enabled by Deep Learning (DL) have emerged as promising techniques for solving various problems in the Internet of Underwater Things (IoUT) [8] and equally in aquaculture [9, 10]. In particular, image analysis can be used to identify prawn species, detect prawns in images, measure prawn length, and estimate prawn weight [11]. However, existing methods have some limitations, such as requiring high-quality images with uniform backgrounds, relying on hand-crafted features that may not capture complex variations, or using simple regression models that may not generalize well to different conditions [12]. Moreover, most of these methods do not consider the morphological characteristics of prawns that affect their weight distribution.

In this paper, we propose a novel Deep Learning [13, 14] approach for automated morphometric analyses and weight estimation of prawns from digital images. Our approach consists of

two main components: a Kronecker product-based feature extraction module (KPFEM), and a landmark localization module (LLM). The KPFEM uses the Kronecker product operation [15] to combine low-level and high-level features from different convolutional layers efficiently. The LLM predicts the coordinates of key points on the prawn body using a localization network. For weight estimations, we have designed a weight regression module (WRM) that works based on the extracted landmarks using a fully connected network. We also use the landmarks generated by the LLM component to perform morphometric analysis. To the best of our knowledge, this is the first work that applies the Kronecker product operation for feature extraction in morphometrics analysis. Moreover, this is the first work that uses a localization network for landmark detection in prawn images.

The main contributions of our paper are as follows:

1) We introduce a novel approach to prawn image analysis in aquaculture settings.
2) Our method incorporates a unique feature extraction technique.
3) We apply a deep network for landmark localization.
4) We perform morphometric shape analyses and use Principal Component Analysis (PCA) to find correlations.
5) We design a weight regression model.
6) Our approach has been evaluated on a large dataset of prawn images.

2 Material and Methods

Our proposed deep learning architecture is shown in Fig. 1 with two distinct outputs, one for prawn weight, and the second for landmark identification and applied morphometric analyses.

The KPFEM serves as a feature extraction module and utilizes Kronecker convolution operation in a novel network architecture to extract features from the input prawn image. The resulting feature map is then passed to the LLM, which uses a deep learning-based approach to detect 12 landmarks on the prawn body.

The predicted landmarks are then used to calculate the distances between any 12 landmarks, resulting in a total of $(12(12-1)/2 = 66)$ possible distances. These distances are then fed to the WRM, which uses a deep learning-based approach to predict the weight of the prawn. The 12

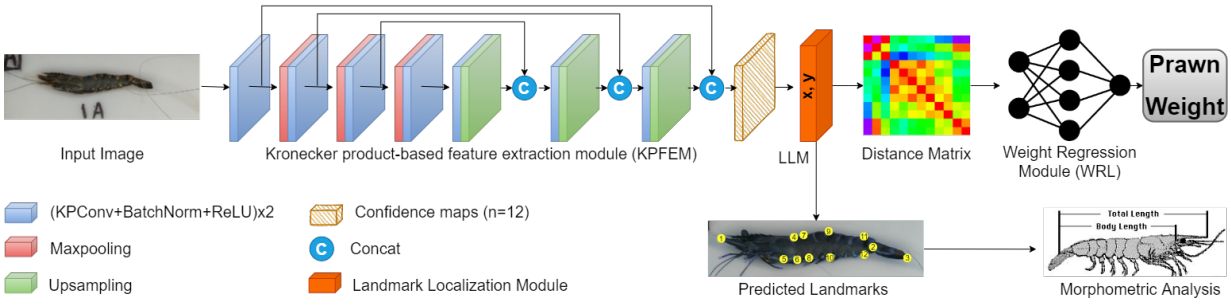


Fig. 1. An overview of our proposed network architecture used for landmark detection from images that can be used for morphometric analyses and weight estimation. Our architecture consists of two main modules: a Kronecker product-based feature extraction module (KPFEM) and a subsequent landmark localization module (LLM). These are followed by a weight regression module (WRM). The KPFEM module is responsible for extracting features from the input image using Kronecker product-based convolutional layers, while the LLM module localizes the landmarks of the prawn using the extracted features. The WRM module regresses the weight of the prawn based on the detected landmarks. This multi-stage approach has shown promising results for accurate morphometric estimation and prawn weight from images.

detected landmarks are also used to perform morphometric analysis of the prawn. The following subsections provide further details on the role and importance of each of the aforementioned modules in reaching the overall goal of our architecture, i.e. automatic morphometric analysis and weight estimation from prawn images.

2.1 Kronecker product-based feature extraction module (KPFEM)

2.1.1 Kronecker Convolution Layer

The Kronecker convolution operation is based on the Kronecker product which is a mathematical operation that takes two matrices and produces a new matrix that is formed by multiplying each element of the first matrix with the second matrix. In the context of convolutional neural networks, the Kronecker product is used to create a weight tensor that is a Kronecker product of two smaller tensors, one that represents the filters in the convolutional layer and another that represents the input image, to create a large weight tensor. This weight tensor is then used to compute the convolution between the input image and the filter.

In contrast to standard convolutional layers, which apply a single filter across all input channels to find local spatial relationships between input features, the Kronecker convolution operation applies a filter that captures both spatial and higher-order information across all input

channels. This makes it an effective approach for feature extraction in multi-channel inputs such as RGB images.

The Kronecker product can be written as follows:

$$\underset{(n\times n)}{[\mathbf{A}_1]} \otimes \underset{\left(\frac{s}{n}\times\frac{d}{n}\times k\times k\right)}{\left[\ \mathbf{F}_1\ \right]} + \ldots + \underset{(n\times n)}{[\mathbf{A}_n]} \otimes \underset{\left(\frac{s}{n}\times\frac{d}{n}\times k\times k\right)}{\left[\ \mathbf{F}_n\ \right]} = \underset{(s\times d\times k\times k)}{\begin{bmatrix} \\ \\ \\ \mathbf{H} \\ \\ \\ \end{bmatrix}}. \tag{1}$$

where matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ and the filter matrix $\mathbf{F}$ and the weight matrix $\mathbf{H}$ have the same dimensions: s channels, d filters, and $k \times k$ kernel size.

The Kronecker Convolution Layer (KCL) uses the Kronecker product to arrange convolution filters in a way that reduces the number of parameters by a factor of $1/n$. We explain how this works for different values of n in Eq. 1. When $n = 1$, we have a real-valued convolution and the Kronecker product is just a scalar multiplication. The filter matrix $\mathbf{F}$ has the same size as the weight matrix $\mathbf{H}$, which is $s \times d \times k \times k$.

When $n = 2$, we have a complex-valued convolution and the Kronecker product is between two matrices. The filter matrices $\mathbf{F}_1$ and $\mathbf{F}_2$ are half the size of $\mathbf{H}$, and they contain the filters for each complex component. The algebra is done with matrices $\mathbf{A}_1$ and $\mathbf{A}_2$. This way, we use half as many parameters as in the real case. When $n > 2$, we can extend this idea by using smaller filter matrices for each dimension. The size of $\mathbf{H}$ does not change, but the parameter size decreases with higher values of n.

The weight tensor $\mathbf{H}$ in the KCL is obtained by summing Kronecker products between two groups of learnable matrices. Specifically, it can be expressed as:

$$\mathbf{H} = \sum_{i=1}^{n} \mathbf{A}_i \otimes \mathbf{F}_i, \tag{2}$$

where $\mathbf{A}_i$ is a $n \times n$ matrix that describes the algebra rules, and $\mathbf{F}_i$ is a $\frac{s}{n} \times \frac{d}{n} \times k \times k$ matrix

representing the i-th batch of filters. These filters are arranged according to the algebra rules to construct the final weight matrix.

Algorithm 1 is a pseudocode implementation of Kronecker Product Convolution using PyTorch-like syntax. The algorithm takes two input tensors: A, a 2D tensor of shape (height, width), and F, a 4D tensor of shape ($num\_filters$, $num\_channels$, $filter\_height$, $filter\_width$), and performs Kronecker product convolution on them. The output tensor has a shape of ($num\_filters$, $num\_channels$, $output\_height$, $output\_width$), where $output\_height$ and $output\_width$ are calculated based on the size of A and the filter size.

The algorithm works by first computing the Kronecker product of A and F, which is a block matrix of shape ($num\_filters$, $num\_channels$, $filter\_height$, $filter\_width$). This is achieved by expanding the dimensions of A and F and multiplying them element-wise. The resulting block matrix is then reshaped to the desired output shape.

Algorithm 1: Kronecker Product, PyTorch-like

```
import torch

def kronecker_product(self, A, F):
    mtx1 = torch.Size("torch.tensor"(A.shape[-2:]) * "torch.tensor"(F.shape[-4:-2]))
    mtx2 = torch.Size("torch.tensor"(F.shape[-2:]))
    res = A.unsqueeze(-1).unsqueeze(-3).unsqueeze(-1).unsqueeze(-1) *
    F.unsqueeze(-4).unsqueeze(-6)
    mtx0 = res.shape[:1]
    out = res.reshape(mtx0 + mtx1 + mtx2)
    return out
```


2.1.2 Standard Convolutional Layer

A standard convolutional layer convolves the input $\mathbf{x} \in \mathbb{R}^{t \times s}$ with the filter tensor $\mathbf{W} \in \mathbb{R}^{s \times d \times k \times k}$ to generate the output $\mathbf{y} \in \mathbb{R}^{d \times t}$, as follows:

$$\mathbf{y} = \text{Conv}(\mathbf{x}) = \mathbf{W} * \mathbf{x} + \mathbf{b}, \tag{3}$$

where s is the input channels dimension, d the output, k is the filter size, and t is the input and output dimension. The bias term $\mathbf{b}$ has negligible impact on the number of parameters, resulting in a complexity of $\mathcal{O}(sdk^2)$.

The KCL is a convolutional layer that uses a weight tensor $\mathbf{H}$ to organize its filters, which is constructed by summing Kronecker products. The layer can be defined as

$$\mathbf{y} = \text{KCL}(\mathbf{x}) = \mathbf{H} * \mathbf{x} + \mathbf{b}, \tag{4}$$

where $\mathbf{H}$ is a learnable tensor with dimensions $s \times d \times k \times k$. The two groups of learnable matrices used to construct $\mathbf{H}$ are denoted as $\mathbf{A}_n$ and $\mathbf{F}_n$, which are combined through Kronecker products to create $\mathbf{H}$, (see Eq. (1) and Eq. (2)). The value of n can be set by the user to specify the real or hypercomplex domain, and controls the degree of freedom of $\mathbf{A}_n$ and $\mathbf{F}_n$. The number of parameters in the KCL is reduced by a factor of $1/n$ compared to a standard convolutional layer in real-world problems, because typically $sdk^2 \gg n^3$. During training, the matrices $\mathbf{A}_n$ and $\mathbf{F}_n$ are learned and used to construct $\mathbf{H}$. The dimensions of $\mathbf{F}_n$ are $\frac{s}{n} \times \frac{d}{n} \times k \times k$ for squared kernels, and $\frac{s}{n} \times \frac{d}{n} \times k$ for 1D kernels. Hence, The KLC complexity of the weight matrix can be approximated to $\mathcal{O}(sdk^2/n)$.

2.1.3 KCL advantages compared to standard convolution

Compared to standard convolutional layers, using KCL brings several advantages. As discussed, firstly, the KCL reduces the number of parameters by a factor of $1/n$ in real-world problems, where n is the hyperparameter that specifies the desired domain. For example, for RGB images that have $n = 3$, the network number of parameters is reduced by 66%. This reduction in parameters can lead to faster training and inference times, as well as reduced memory usage, making KCL well-suited for resource-constrained devices. Secondly, KCL allows for weight sharing among different channels in multidimensional data, such as colour images, which enables capturing latent intra-channel relations that standard convolutional networks may ignore due to the fixed structure of the weights. This can result in better performance in tasks that involve correlated channels. Finally, the KCL can be easily integrated into any convolutional model by replacing standard convolution or transposed convolution operations, and the hyperparameter n provides high flexibility to adapt the layer to any kind of input. Overall, the KCL offers a promising alternative to standard convolutional layers and has the potential to improve the performance of convolutional neural networks in various applications.

2.1.4 Feature Extraction Structure

The proposed KPFEM module is composed of 14 KCLs that extract features from the input prawn image. As shown in Fig. 1 each layer is followed by a rectified linear unit (ReLU) activation function and every second layer is followed by a max pooling operation to reduce the spatial dimension of the feature maps. The module has ”skip connections”, which allow information to flow through the network more efficiently by skipping over certain layers that might otherwise impede the flow of information.

2.2 Landmark Localization Module (LLM)

The landmark localization module takes the extracted features generated by the KPFEM module and predicts the locations of the landmarks. Therefore, for our keypoint detection task, instead of using Fully Convolutional Network (FCN) to directly predict a numerical value of each keypoint coordinate as an output (i.e. regressing images to coordinate values), we modified FCN to predict a stack of output heatmaps (i.e. confidence maps), one for each keypoint. The position of each keypoint is indicated by a single, two-dimensional, symmetric Gaussian in each heatmap in the output, and the scalar value of the peak reflects the prediction's confidence score.

Moreover, our proposed LLM not only predicts heatmaps but also predicts scalar values for coordinates of each keypoint. Therefore, during the training process, we have a multi-task loss function, which consists of two losses, i.e. Jensen-Shannon divergence for heatmaps and Euclidean distance for coordinates. The first loss measures the distances between the predicted heatmaps and the ground-truth heatmaps, while the second loss measures the distances between the predicted coordinates and the ground-truth coordinates. Then, we take the average of the two losses as the optimization loss. As demonstrated in Fig. 1, the LLM output is a set of predicted landmark coordinates, using which also a distance matrix is produced to feed to the next module in our proposed architecture.

The accuracy of measurements is critical for the quantitative comparison of prawn morphometry, where accuracy refers to the closeness of measurements to the true value. To assess accuracy, this study utilized the mean absolute difference (MAD) between manual and DL measurements. The MAD is a measure of accuracy that calculates the average absolute deviation between two

values. It is obtained by dividing the sum of absolute deviations between each value by the number of values. The mathematical expression for MAD is:

$$MAD = \frac{1}{n}\sum_{i=1}^{n}|(x_i - y_i)|$$

where n is the total number of observations, x_i and y_i are the values of the $i-th$ observation in two different samples, and the vertical bars denote absolute value.

2.3 Weight Regression Module (WRM)

The final component of our architecture is the weight regression module which is made up of a multilayer perceptron (MLP) that consists of five layers of nodes: an input layer, three hidden layers, and an output layer. Each node in the hidden layers applies the ReLU activation function to the weighted sum of inputs from the previous layer.

The weight regression module takes the output of the landmark localization module, which includes the predicted locations of the landmarks, and measures 66 distances between any 12 landmarks to predict the weight. We use the distances between the landmarks to estimate significant traits such as total length, body length, carapace length, and length-width ratios. These measurements are typically made between easily distinguishable landmarks. The distances between them are assigned a trait name and then used as inputs for a pre-trained regression model that maps these estimates to the prawn weight. Specifically, we generated morphological measurements using a combination of methods from previous studies and those that were possible within the constraints of the available images. For the morphometric shape analysis, we used 8,164 photographed specimens to estimate 66 morphometric distances derived from 12 landmarks. These distances were then used by the weight regression module to predict the prawn weight directly from the landmarks. This is a practical and useful application of our model. We trained our model using a mean squared error (MSE) loss to minimize the difference between the predicted weight and the ground-truth weight. The ground-truth weight of the prawns was measured using a digital scale with a precision of $0.01g$. Each prawn was carefully placed on the scale and the weight was recorded.

In the shape or contour features method for prawn weight estimation, Sec. 2.4, the prawn body is segmented from the image using a threshold segmentation process. This process involves

selecting a threshold value that separates the pixels in the image into two groups: foreground and background. The pixels with intensity values above the threshold are classified as foreground pixels, which belong to the prawn body, while the pixels with intensity values below the threshold are classified as background pixels. This is demonstrated in Fig. 3.

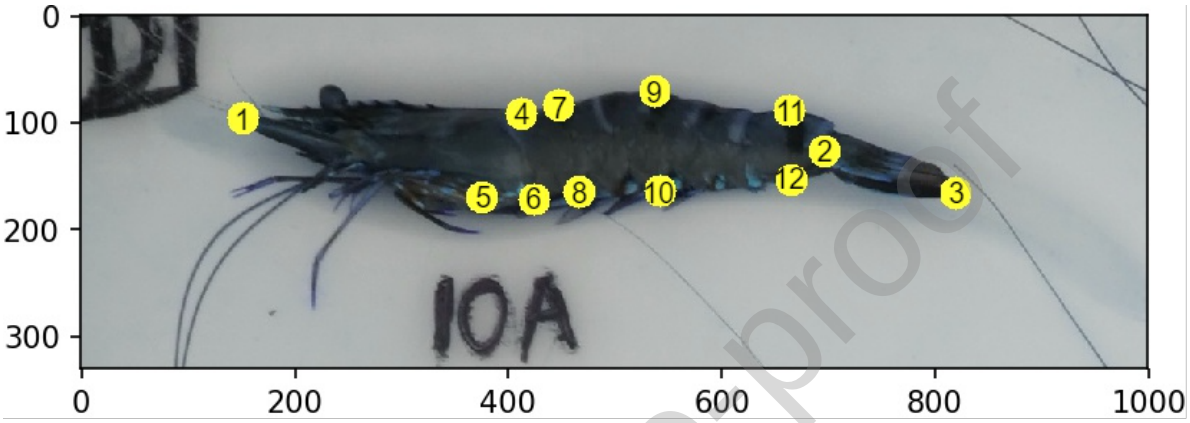


Fig. 2. shows the landmarks used in our analysis marked on the body of the Black Tiger Prawn (*Penaeus monodon*): (1) most anterior point of the antennal scale, (2) most anterior point of the tail, (3) most posterior point of the tail, (4) junction of the carapace and abdomen at the most dorsal point, (5) midway along the carapace on the ventral side of the prawn, (6) junction of the carapace and abdomen at the most ventral point, (7) dorsally on the midpoint of the first abdominal segment, (8) ventrally on the midpoint of the first abdominal segment, (9) dorsally on the midpoint of the third abdominal segment, (10) ventrally on the midpoint of the third abdominal segment, (11) dorsally on the midpoint of the last abdominal segment, and (12) ventrally on the midpoint of the last abdominal segment.

2.4 The Proposed Weight Estimation

We compared our proposed approach with two existing methods for prawn weight estimation: the traditional linear regression method [16], and a Deep Learning-based method [17] based on shape or contour features for weight estimation. The prawn weight estimation method uses threshold segmentation to separate the prawn body (foreground) from the image background based on pixel intensity. This process is shown in Fig. 3.

After segmenting the prawn body, the next step is to estimate its weight. One way to do this is to count the number of pixels in the segmented region, which is assumed to be proportional to the prawn weight. This pixel count is then correlated to the actual weight of the prawn using two approaches: 1) Linear Regression Model: This method uses a mathematical model to correlate the pixel count (obtained from segmenting the prawn body) with the actual weight of the prawn. The

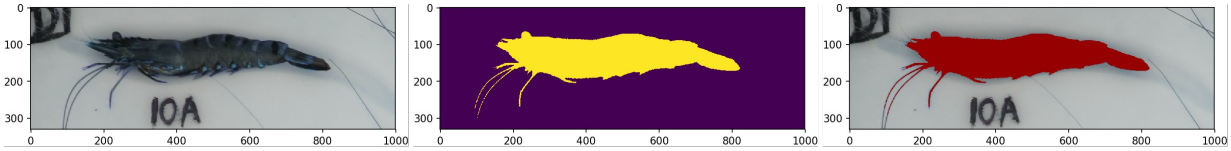


Fig. 3. Example of threshold segmentation. From left to right: (a) original image, (b) segmented prawn body, (c) overlay of the segmentation on the original image.

assumption here is that the number of pixels is proportional to the prawn's weight. This method is similar to the one used in [16] for fish weight estimation.

2) Deep Learning Model: This method uses a neural network that predicts the prawn's weight based on the pixel count. The neural network is trained to recognize patterns and make predictions based on the input data (pixel count in this case). This method is similar to the one used in [17].

The key difference between these two methods lies in their approach, while the linear regression model assumes a direct proportionality between pixel count and weight, the deep learning model can capture more complex relationships between these variables. However, these methods have some limitations. For example, it assumes that the relationship between pixel count and prawn weight is linear. Additionally, this method does not take into account the shape and position of the prawn body, which can vary from one image to another.

These methods were compared against our proposed approach for prawn weight estimation, which involves generating 66 possible distances from the 12 landmarks and feeding them into a weight regression module. We generated a distance matrix as seen in Fig. 4 using the 12 predicted landmarks. This distance matrix contains the pairwise Euclidean distances between all possible pairs of the 12 landmarks. The number of possible distances between 12 landmarks can be calculated using the combination formula, which is given by: $C(n,k) = \frac{n!}{k!(n-k)!}$ where, n is the total number of landmarks (12 in this case), and k is the number of landmarks to be chosen at a time (2 in this case, because distance is between two points). Substituting $n = 12$ and $k = 2$ into the formula: $C(12,2) = \frac{12!}{2!(12-2)!} = \frac{12!}{2!10!}$ This simplifies to: $C(12,2) = \frac{12\times11}{2} = 66$ possible distances. These distances are then fed to the WRM, which uses a deep learning-based approach to predict the weight of the prawn.

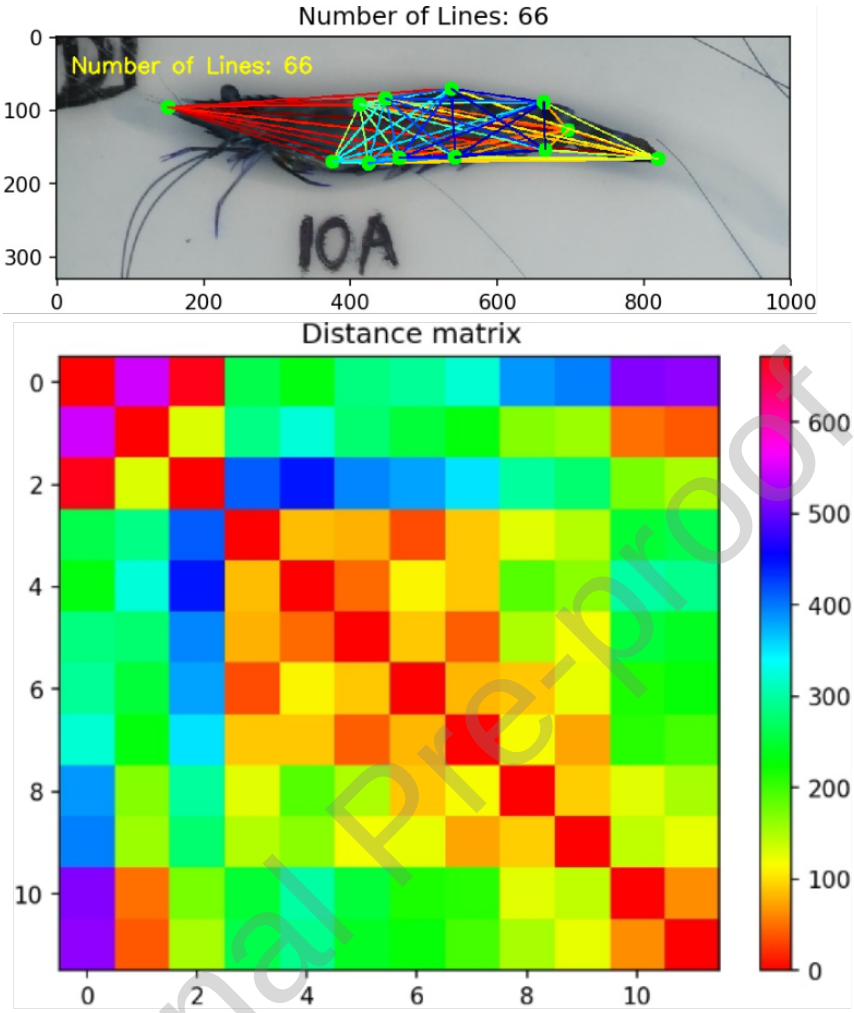


Fig. 4. Top: Image of a prawn with the 66 possible distances between its 12 landmarks marked. Bottom: The resulting distance matrix plot computed from these distances.

2.5 Principal Component Analysis (PCA)

In addition to our aforementioned analysis, we also performed the PCA on the predicted landmarks from our model to analyze the morphometric shape of prawns. We used PCA as a data-driven approach to analyze prawn shape variation. Despite having ground truth measurements, PCA helped us understand the major sources of shape variation in prawns, providing a holistic view beyond individual distance measurements. We acknowledge that Principal Components (PCs), derived from the covariance matrix of our data, are mathematical constructs. They represent the directions of maximum variance in the data. While they are not directly tied to any specific

physical or biological characteristics of the prawns, analyzing these PCs allows us to gain insights into the major sources of variation in prawn shape. This understanding can be practically useful in various ways. For instance, it can help in identifying key features that differentiate between different prawn species or in understanding how prawn shape changes with growth or environmental conditions. We believe this clarification accurately represents the role and interpretation of PCs in our analysis.

The cos2 values demonstrate the quality of representation of the individuals by the principal components. In our PCA analysis, the term 'cos2' refers to the square of the cosine of the angles between the data points and the principal components. These 'cos2' values are used to assess the quality of representation of the individuals by the principal components. High 'cos2' values indicate that the individuals are well represented by the principal components.

It is important to note that the results obtained from the PCA are based on the predicted landmark data, which may have some inaccuracies. However, evidence points to the predicted landmark data still provide valuable insights into the shape variation among prawns and can be used as a proxy for understanding the morphometric characteristics of the prawns. This is especially important because collecting manual landmarks is impractical at large scales.

3 EXPERIMENTS

3.1 The Dataset

We collected images of 8164 individual Black Tiger Prawns (*Penaeus monodon*) from an aquaculture farm in Australia. The images were captured using a digital camera with a resolution of 12 megapixels, a sensor size of $1/2.3$", and a lens focal length of $24 - 70mm$. The camera was set to automatic mode to ensure optimal lighting and focus. The images were captured under natural lighting and saved as JPG files. We used ImageJ software to annotate each image with 12 landmarks that correspond to prawn size and shape (Fig. 2). The landmark positions were defined based on a previous study [18] and adapted to our image quality.

To evaluate the reliability of the landmark data, we measured 300 images twice by one operator (O1) and once by another operator (O2) and computed the intraclass correlation coefficient (ICC) for each landmark pair using R software. The ICC values ranged from 0.96 to

0.99, indicating high agreement between operators. The annotated images contain 12 specific and homologous points on the prawn body. For a complete list of the 12 landmarks, see Fig. 2. We then derived 5 traits from these landmarks, which are listed in Table 1 and used for morphometric analyses in Sec. 4.4. The measurements in Table 1, were chosen for their relevance to prawn morphometrics and their ability to provide a comprehensive understanding of prawn size and shape. Typically, all measurements in morphometric analyses are considered equally important to avoid bias. These specific measurements were selected based on their biological significance and potential variability among prawns. The choice of other measurements can depend on the research question or objective, with these measurements being particularly relevant for recognizing the relative size of prawns.

The dataset, therefore, includes for each image, the 12 ground-truth landmark coordinates, and morphometric measurements obtained from the annotated coordinates on images. The dataset was divided into three parts: training, validation, and testing. We used the training dataset to train the Deep Learning models, the validation dataset to assess model performance during training, and the testing dataset to evaluate the performance of the trained models. The ground truth landmarks for each prawn image had the form $[(x_1, y_1), .., (x_k, y_k)]$, where (x_i, y_i) represented the ith landmark location. Our model was then trained to predict keypoint locations for each prawn image as shown in Fig. 2. The predicted landmarks had the same form as the ground truth.

We carefully curated the dataset to ensure a balanced representation of different prawn weights and body shapes, to enhance the robustness of the results. We also pre-processed the images to make sure they were of similar size and resolution and to eliminate background noise and other distractions that could impact the accuracy of landmark identification. That preprocessing will be described in greater detail later in Sec. 3.3.1.

3.2 Evaluation Metrics

In our study, we used several metrics to compare the performance of different models. One of these metrics is the number of Floating Point Operations (FLOPs). FLOPs is a measure of computer performance, useful in fields of scientific computations that require floating-point calculations. It represents the number of operations involving floating-point numbers that a model or algorithm performs. Another metric used in this study is the network throughput in images per

second. It is worth noting that the actual throughput of a network is the number of instances it can process in one second with the optimal batch size. It is obtained by dividing the total number of instances processed by the total time taken to process them and can vary depending on various factors such as the model's complexity, the input data size, and the available hardware resources. Specifically, in this section, we describe the performance metrics used to optimize and evaluate the model and compare the quality of the predicted keypoint locations.

3.2.1 Euclidean distance

The first metric is the Euclidean distance, which measures the distance of the landmarks based on their coordinates and does not depend on how the ground truth has been determined. A value of 0 indicates that the predicted keypoint is exactly at the same coordinate as the ground truth keypoint. We calculate the sum of the squared Euclidean distance of the difference between two kyepoints, i.e., the predicted keypoint and the ground truth human-annotated keypoint. This represents the total difference between the two points. The equation for Euclidean distance is shown in Equation 5,

$$d(g,p) = \sqrt{\sum_{i=1}^{n}(v_i^g - v_i^p)^2}, \tag{5}$$

where g and p are two sets of points in Euclidean n-space for ground truth and prediction, respectively, v_i^g, v_i^p are Euclidean vectors starting from the origin of the space (initial point) for the ground truth and prediction, respectively, and n is the number of landmarks.

3.2.2 Jensen-Shannon divergence

The second metric is the Jensen-Shannon divergence, which is a distance measure between two distributions and can be used to quantify the accuracy of the predicted landmarks distribution compared to the ground-truth landmarks distribution. The lower the value, the better the model performs. This distance is calculated based on the Kullback-Leibler divergence (KLD) and can be expressed as shown in Equation 6.

$$KLD(P||Q) = \sum_{i=1}^{n} p_i(x) log\left(\frac{p_i(x)}{q_i(x)}\right), \tag{6}$$

TABLE 1
Important prawn traits, their descriptions, and their corresponding landmark coordinates from Fig. 2

Trait	Landmark Coordinates	Trait Description
Total length	1-3	From the most anterior point of the antennal scale to the most posterior point of the tail
Body length	1-2	From the most anterior point of the antennal scale to the most anterior point of the tail
First abdominal segment height	7-8	From the dorsal midpoint to the ventral midpoint on the first abdominal segment
Third abdominal segment height	9-10	From the dorsal midpoint to the ventral midpoint on the third abdominal segment
Last abdominal segment height	11-12	From the dorsal midpoint to the ventral midpoint on the last abdominal segment

where The KLD for two probability distributions P and Q, and when there are n pairs of predicted p, and ground truth q.

The Jensen-Shannon divergence (JSD) is then calculated using Equation 7,

$$JSD_M(P||Q) = \sqrt{\frac{KLD(p \parallel m) + KLD(q \parallel m)}{2}}, \tag{7}$$

where m is the point-wise mean of p and q. The JSD measures the difference between two probability distributions, with a value of 0 indicating no difference between the distributions.

3.2.3 Object Keypoint Similarity (OKS)

The third metric used to evaluate the performance of our landmark detection model is the Object Keypoint Similarity (OKS), which is determined by dividing the distance between expected and ground truth points by the object's scale. OKS landmarks estimation serves the same purpose as Intersection over Union (IoU) as in object detection. This gives the similarity between the landmarks of the two detected boxes, with a result between 0 and 1, where 0 means no similarity between the landmarks, while perfect predictions will have OKS=1. The equation for OKS is shown in Equation 8.

$$OKS = \frac{\sum_i \exp\left(-d_i^2/2s^2k_i^2\right)\delta\left(v_i > 0\right)}{\sum_i \delta\left(v_i > 0\right)}, \tag{8}$$

where d_i is the Euclidean distance between the detected keypoint and the corresponding ground truth, v_i is the visibility flag of the ground truth, s is the object scale, while k_i represents a per-keypoint constant that controls falloff.

Equations 5 and 7 were used for model training and optimization and also used to compare different models' performance, as shown in Table 2. Equation 8 was used as a final evaluation metric for all the models used in this study, as shown in Table 3.

3.2.4 Precision and Recall

In this Section, we discuss the performance metrics used to evaluate our keypoint detector model. These metrics are based on Object Keypoint Similarity (OKS) [19], which is a widely accepted standard for this type of evaluation.

The OKS-based metrics provide a comprehensive view of our model's performance by considering both precision and recall across different thresholds. Specifically, we use six metrics to characterize our model's performance:

- Average Precision (AP):
  - AP (at $OKS = .50 : .05 : .95$ (primary metric))
  - $AP^{.50}$ (at $OKS = .50$)
  - $AP^{.75}$ (at $OKS = .75$)
- Average Recall (AR):
  - AR (at $OKS = .50 : .05 : .95$)
  - $AR^{.50}$ (at $OKS = .50$)
  - $AR^{.75}$ (at $OKS = .75$)

These metrics were chosen because they allow us to assess our model's ability to detect landmarks accurately (precision) and comprehensively (recall) across a range of OKS thresholds.

This section builds on the previous sections by applying the concepts and methods discussed earlier to the specific task of evaluating our keypoint detector model.

3.3 Model training

3.3.1 Data Preprocessing

Before the training process can begin, the image data must be preprocessed to ensure that it is suitable for input into the Deep Learning model. This includes resizing the original image size of 1000×331, to a consistent size of 320×320, normalizing the pixel values, and converting the images to grayscale if necessary. In addition, the annotated landmark locations must be converted into a format that can be used by the model, such as a heatmap or a set of points. In our study, we applied a series of image transformation operations to the images in our training set. Each operation was applied with a certain probability, meaning that each image had a chance of undergoing that transformation. These transformations are a common technique in deep learning called data augmentation, which helps to increase the diversity of the training data and improve the models ability to generalize. These operations are: Horizontal flip: 0.5, Vertical flip: 0.5, Shift and scale: 0.5 (shift limit = $0.0625°$, scale limit = $0.20°$), Rotation: 0.5 (rotation limit = $20°$), Blur: 0.3 (blur limit = 1), RGB-shift: 0.3 (R-shift limit = 25, G-shift limit = 25, B-shift limit = 25). These operations help to improve the robustness of our model to lighting changes [20]. We did not transform the images in our validation or test sets in any way.

The ground truth landmarks for each prawn image are represented in the form of $[(x_1, y_1), .., (x_k, y_k)]$, where (x_i, y_i) denotes the location of the ith landmark. In the training process, both the original (x_i, y_i) values and the converted heatmap derived from these values are utilized in the loss function.

3.3.2 Model Selection

Once the data has been preprocessed, the next step is to select a model architecture that is suitable for our problem. There are a variety of Deep Learning models that can be used for image landmark identification. In this work, we have selected six models for our experiments: U-net [21], ResNet-18 [22], ShuffleNet-v2 [23], MobileNet-v2 [24], SqueezeNet [25], and our proposed KPFEM.

3.3.3 Hyperparameter Tuning

Once the model architecture has been selected, the next step is to tune the hyperparameters to achieve optimal performance. This includes selecting the optimal batch size, learning rate, and number of epochs. The hyperparameters are selected using a combination of grid search and cross-validation to ensure that the model is generalizing well and not overfitting to the training data. For this problem set, we chose a learning rate of 1×10^{-3} as the best option. All models took about 200 epochs to train on this problem and we reduced the learning rate by $\gamma = 0.001$ after every 50 epochs. We also used Adam optimiser [26] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1.0 \times 10^{-08}$. We applied these hyperparameters to all six models. The best model configuration may vary depending on the application, so these results do not cover all possible model configurations.

We split the dataset into three sets: "Train", "Validation", and "Test", comprising 40%, 20%, and 40% of the data, respectively. We trained all models on the Train subset of the data with the same hyperparameters. All models had two outputs (heatmap and coordinates) with two losses (see Sec. 3.2). All models took 320×320 input images and produced 56×56 output heatmaps except U-net which had 320×320 output images.

3.3.4 Training

The final step in the training process is to train the model using the preprocessed data and the optimized hyperparameters. The model is trained using a supervised learning approach, where the ground truth landmark locations are used to calculate the loss function and update the model parameters. The training process is repeated until the model has reached convergence or a maximum number of epochs has been reached. We used Pytorch framework [27] on a Linux host with a single NVidia GeForce RTX 2080 Ti GPU and a batch size of 64.

Once the model was trained, it was evaluated on the test subset of the dataset to assess its performance in identifying landmarks in new, unseen images. The evaluation metrics, described in the evaluation metrics section (Sec. 3.2), are used to quantify the accuracy of the model and provide insight into its strengths and weaknesses.

4 RESULTS

4.1 Landmark Detection

Table 2 presents a comparative analysis based on several metrics, including the number of floating-point operations (FLOPs), the number of parameters, model size, and throughput in images per second, as well as the coordinates loss (Equ. 5), heatmap loss (Equ. 7), and the average of both losses.

TABLE 2
Landmark Detection Performance Comparison of our model compared to five benchmark models.

Network	FLOPs (x10^6)	#Params (x10^6)	Size (MB)	Throughput (img/sec)	Coords [1]	HeatMap [2]	Avg. [3]
U-net [21]	16.52	31.04	124.3	201	0.024	0.355	0.190
ResNet-18 [22]	2.62	12.85	51.5	404	0.028	0.090	0.059
ShuffleNet-v2 [23]	0.44	3.06	12.5	170	0.047	0.153	0.100
MobileNet-v2 [24]	0.67	4.10	16.7	205	0.041	0.137	0.089
SqueezeNet [25]	0.92	2.33	9.4	551	0.027	0.078	0.052
KPFEM (ours)	0.01	0.39	1.6	562	0.023	0.084	0.0053

The experimental results, presented in Table 2 highlight the superior performance of our proposed network, which is based on our KPFEM feature extraction method. This network outperforms others in several aspects. Firstly, it has the lowest number of GFlops and parameters, with 47 times fewer parameters than U-net [21]. Secondly, it occupies the smallest size on the hard disk. Thirdly, it has the highest throughput. Lastly, our model exhibits a lower average loss than other popular models, including U-net [21], ShuffleNet-v2 [23], and MobileNet-v2 [24].

Table 3 shows the performance of our model on the test subset of the dataset compared to benchmark models in landmark detections, using the OKS evaluation metric. To assess the generalization effectiveness of our model, we compared the performance of our model with randomly initialized weights, against the other models with randomly initialized weights as well to provide a direct comparison. Table 3 demonstrates that our proposed model performs well in

1. This loss corresponds to the coordinates loss (Equ. 5).
2. This loss corresponds to heatmap loss (Equ. 7).
3. This loss corresponds to the average of both losses (Equ. 5 and Equ. 7).

generalization with only 40% of the data and without the use of transfer learning, when combined with robust data augmentation techniques.

The overall outcome depicted in Table 3 indicates that our proposed network surpasses both ShuffleNet-v2 [23] and MobileNet-v2 [24] in terms of landmark detection performance, with an accuracy of $AP = 0.986$, while still competing with SqueezeNet [25] even though it has substantially fewer parameters. It is noteworthy that our model attains this high accuracy with only 0.39M parameters and without relying on transfer learning. These results clearly demonstrate the effectiveness and generalisability of our KPFEM-based model.

TABLE 3
Performance comparison using the OKS metric on the **test** dataset.

Network	AP	$AP^{.50}$	$AP^{.75}$	AR	$AR^{.50}$	$AR^{.75}$
U-net [21]	0.981	0.990	0.990	0.974	0.999	0.999
ResNet-18 [22]	0.984	0.990	0.990	0.982	0.999	0.999
ShuffleNet-v2 [23]	0.957	0.990	0.990	0.979	0.999	0.999
MobileNet-v2 [24]	0.963	0.990	0.989	0.979	0.999	0.996
SqueezeNet [25]	0.971	0.990	0.990	0.983	0.999	0.999
KPFEM (ours)	0.986	0.990	0.990	0.985	0.999	0.999

4.2 Weight Estimation

Table 4 shows the comparison results of prawn weight estimation methods. Our proposed approach achieved the lowest mean absolute error (MAE) and mean squared error (MSE) values and the highest coefficient of determination among the three methods. Specifically, our approach achieved an MAE of 0.649 g, an MSE of 0.986 g, and a coefficient of determination of 0.934, which outperformed the other two methods. These results demonstrate the effectiveness and superiority of our proposed approach for prawn weight estimation.

In addition to evaluating the overall performance of our proposed weight estimation approach, we also used visualizations to further understand our model's performance and identify areas for improvement. Specifically, we used Fig. 5 to visualize the relationships between the predicted and

TABLE 4
Comparison of prawn weight estimation methods using mean absolute error (MAE) in grams, mean squared error (MSE) in grams, and coefficient of determination (R^2)

Method	**MAE (g)**	**MSE (g)**	**Coefficient of determination**
Linear Regression	0.893	1.848	0.825
Deep Learning-based Method	0.880	1.713	0.896
Proposed Approach	**0.630**	**0.735**	**0.952**

the true weight values. The plot in Fig. 5 shows the relationship between predicted weight and true weight for three different methods: Linear Regression, Deep Learning-based Method, and the Proposed Approach. The plot can provide information about the accuracy and precision of the different methods. For example, if the points on the plot fall close to the diagonal line ($y = x$), then the predicted weights are close to the true weights, indicating a high level of accuracy. Additionally, if the points are tightly clustered around the diagonal line, then the method is precise in its predictions.

Based on the plots in Fig. 5, it appears that the Proposed Approach has a higher correlation between predicted and true weight values compared to the Linear Regression and Deep Learning-based methods. This can be seen by the tighter clustering of points around the line of best fit. The presence of outliers in the Linear Regression and Deep Learning-based methods may be affecting the overall correlation.

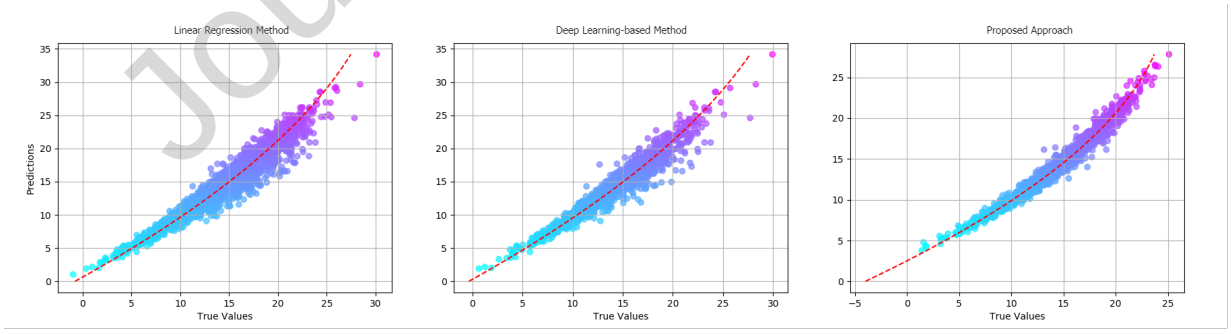


Fig. 5. Plot showing the relationship between predicted weight and true weight for three different methods. From left to right: results for the Linear Regression method, Deep Learning-based Method, and our Proposed Approach.

4.3 Ablation Study on Weight Estimation

In our study, we explored the effect of dimensionality reduction on our proposed method by conducting an ablation study using Principal Component Analysis (PCA). The aim of employing PCA was to condense the large number of features per observation, specifically the 66 potential distances between the 12 landmarks identified on a prawn. These distances were calculated using the combination formula. We implemented PCA on the distance matrix to reduce the dimensionality of our data. The decision on the number of components to retain was guided by the cumulative percentage of explained variance, ensuring an informed, non-arbitrary selection.

For our study, we further examined the impact of reducing the number of components to specific values 2, 5, and 10. These values were chosen to provide a comparative analysis of the model's performance with varying numbers of components. We then used these reduced feature sets to predict prawn weight. The results of this analysis are presented in Table 5.

We found that using the full set of 66 distances without the PCA resulted in better weight estimation results. This suggests that the PCA was not necessary for this particular task and that the high-dimensional feature space was important for accurately capturing the information needed for prawn weight estimation. This finding is also in agreement with the observation shown in Table 5, which demonstrates the more features used, the lower the MAE and MSE of the model.

TABLE 5
Comparison of weight estimation results using the PCA with a different number of components.

Method	MAE (g)	MSE (g)	Coefficient of determination
PCA ($n = 2$)	0.784	0.851	0.915
PCA ($n = 5$)	0.765	0.842	0.924
PCA ($n = 10$)	0.724	0.816	0.931
No PCA	**0.630**	**0.735**	**0.952**

4.4 Morphometric Analyses

In our proposed deep learning model, we detect landmarks on the prawn body to achieve highly accurate weight estimations. By detecting and analysing these landmarks, other important physical characteristics of the prawn, such as its length, width, and shape, can be extracted.

We conducted an analysis based on the evaluation of landmarks, which allowed us to derive important traits for prawn morphometrics. These traits are shown in Table 1, and include total length, body length, the first abdominal segment height (First ASH), the third abdominal segment height (Third ASH), and the last abdominal segment height (Last ASH). We used the landmark data obtained from our model to calculate these five traits for each individual prawn and plotted their distributions in Fig. 6. We also computed the correlation matrix among the traits and visualized it as a heatmap in Fig. 7. The results show that total length and body length are highly correlated ($r = 0.99$), and so are the First and Third, and Third and Last ASH ($r = 0.93$). The other traits had high correlations as well. This coordinated variation in traits represents the diversity in overall shape and size among the prawns in our study.

Table 6 presents a performance comparison of various deep learning (DL) networks in terms of their ability to accurately measure morphometric features. The table shows the mean absolute difference (MAD) in millimetres between manual and DL measurements for total length, body length, first ASH, third ASH, and last ASH. The results indicate that our proposed KPFEM-based network outperforms other networks such as U-net, ResNet-18, ShuffleNet-v2, MobileNet-v2, and SqueezeNet in terms of MAD for all morphometric features. This suggests that our approach is more accurate in measuring morphometric features compared to other DL networks used for landmark detection.

TABLE 6
Morphometric Analyses Performance Comparison: Mean Absolute Difference (MAD) between Manual and DL Measurements (mm)

Network	Total length	Body length	First ASH	Third ASH	Last ASH
U-net [21]	1.81	1.83	1.73	1.72	1.51
ResNet-18 [22]	1.72	1.85	1.71	1.64	1.43
ShuffleNet-v2 [23]	2.34	2.22	2.15	2.15	2.02
MobileNet-v2 [24]	2.23	2.24	2.04	2.12	1.94
SqueezeNet [25]	2.14	1.93	1.94	1.85	1.74
KPFEM (ours)	1.61	1.54	1.53	1.45	1.31

4.5 PCA Analysis

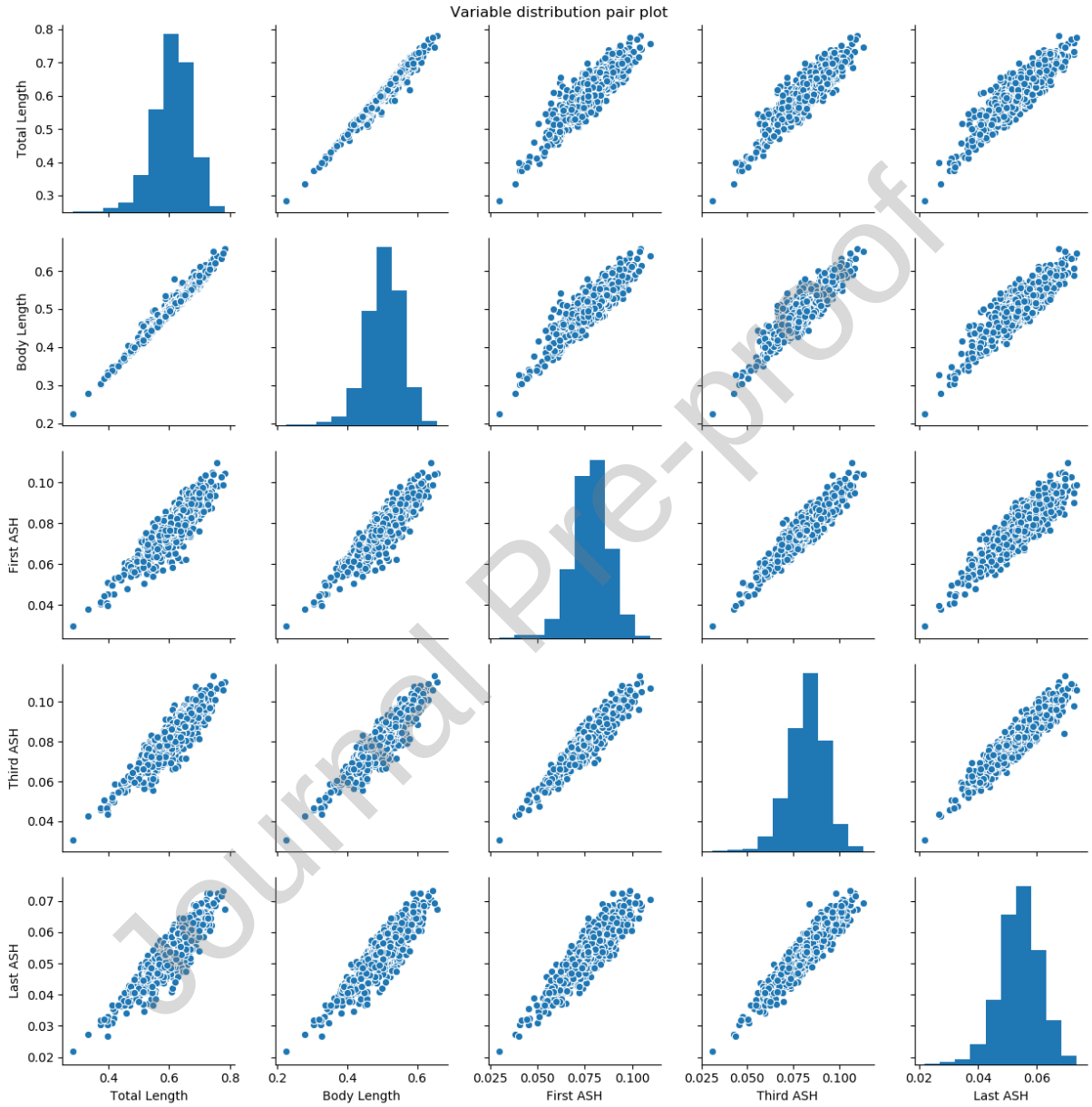


Fig. 6. The distribution pair plots for the five important prawn traits from Table 1, i.e. Total length, Body length, First abdominal segment height ”First ASH”, Third abdominal segment height ”Third ASH”, Last abdominal segment height ”Last ASH”

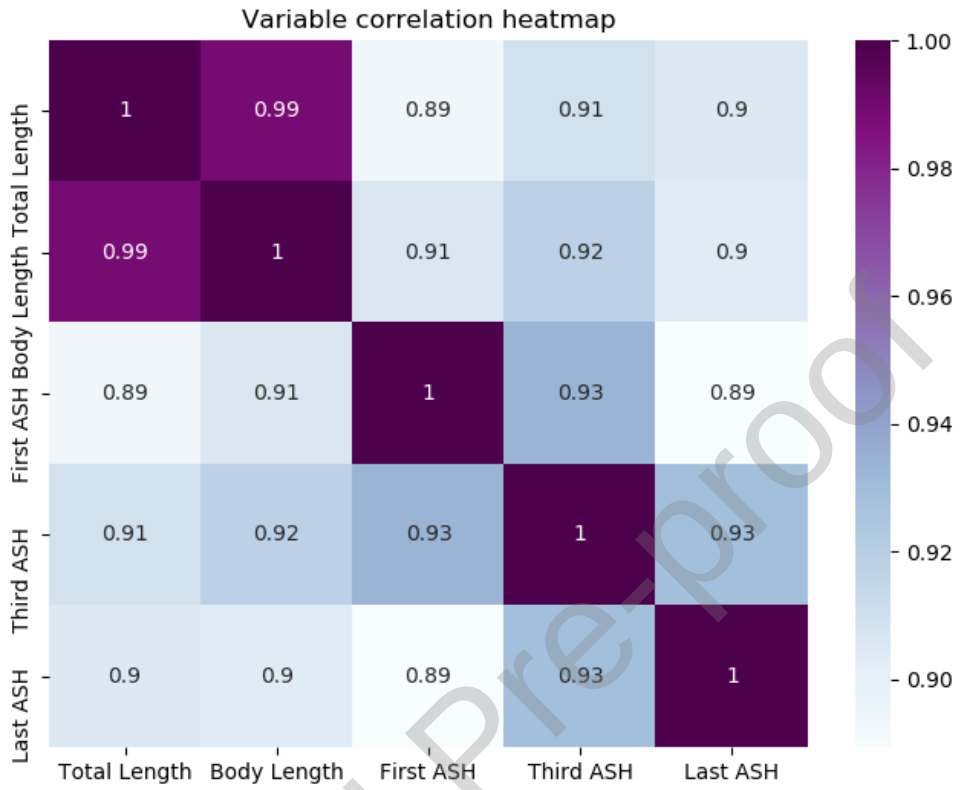


Fig. 7. The correlation heatmap for the five important prawn traits from Table 1, i.e. Total length, Body length, First abdominal segment height "First ASH", Third abdominal segment height "Third ASH", Last abdominal segment height "Last ASH"

In our PCA analysis, we found that the first principal component (PC1) accounted for 91.6% of the total variability in the data. While PC1 accounted for the majority of the variance, we also considered PC2, which accounted for an additional 2.6% of the variability. The decision to include specific components in our analysis was based on their interpretability. Despite the variance accounted for by PC2 being similar to that of subsequent PCs, it was included in our analysis to ensure a comprehensive representation of our data. The detailed results of variability are presented in Table 7.

The scores and vectors for the first two PCs are also shown in Fig. 8. In the left panel, the circles represent individual prawns with their dataset ID shown next to them.

TABLE 7
Explanation of Variability by Various Principal Components (PCs)

	PC1	PC2	PC3	PC4	PC5	PC6
Standard deviation	191.24	23.63	19.22	14.78	11.42	10.17
Proportion of Variance	0.916	0.026	0.014	0.006	0.004	0.002
Cumulative Proportion	0.916	0.942	0.956	0.962	0.966	0.968

In our PCA analysis, PC1 accounted for the majority of the variation (91.6%) among the landmarks. The positive loadings of PC1 suggest that an increase in PC1 scores corresponds to an increase in the distance between landmarks 1 and 3, which is indicative of the total length of the prawn. Therefore, based on the interpretation of these factor loadings, PC1 can be seen as an indicator of overall size. The high variance accounted for by PC1 is likely due to the wide range of sizes present in our samples.

PC2, also referred to as Dim2, explained a smaller portion of the total variation among landmarks (2.6%). The extremes on the PC2 axis, as shown in Fig. 8 (left), correspond to variations in body shape: a very thin/elongated body shape at the low end and a thick/fatty shape at the high end. Thus, based on the interpretation of the factor loadings, PC2 can be seen as an indicator of the proportionate body width/shape of the prawns.

While PC2, which represents the proportionate body width/shape of the prawns, explains a smaller proportion of the total variance, it nonetheless offers valuable insights into shape variations among prawns. However, given its lesser contribution to the total variance, interpretations related to PC2 should be made with caution.

Figure 8)(right) illustrates how different variables (distances) contribute to the variation in shape among prawns with respect to the two main PCs, i.e. PC1 and PC2. Here, each of the arrows shows one of the distances, e.g. d_1_5 designates the distance between landmark 1 and 5, and how it relates to the two PCs.

In conclusion, our PCA analysis of the predicted landmark data revealed that PC1 represents the overall size and PC2 represents the proportionate body width/shape of the prawns. These findings provide important insights into our targeted morphometric characteristics of prawns and lay a foundation for further research on the genetic and environmental factors that affect prawn

morphology [28].

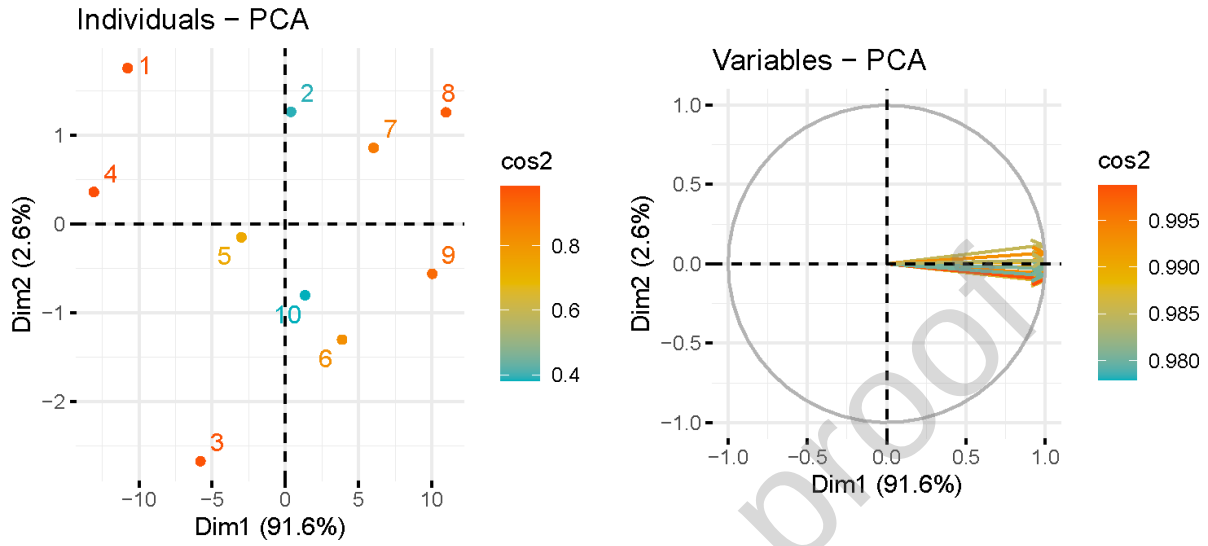


Fig. 8. PCA Analysis. In the left panel, the circles represent individual prawns with their dataset ID shown next to them. Their positions on the plot are determined by their scores on PC1 (Dim1) and PC2 (Dim2). The cos2 values measure the quality of representation of the individuals by the principal components. The right panel illustrates how different variables (distances) contribute to the variation in shape among prawns with respect to the two main PCs. Here, each of the arrows shows one of the distances, e.g. d_1_5 designates the distance between landmark 1 and 5, and how it relates to the two PCs.

5 Discussion

The accurate estimation of weight and other morphological features of farmed individuals in aquaculture is a critical aspect of industrial-scale operations. It aids in improving crop management, supports decision-making processes, and can be integrated into product grading and processing activities. Traditional methods for acquiring such data are labour-intensive, underscoring the need for automated methods. In response to this need, we have proposed a novel approach that takes advantage of computer vision and deep learning. Our approach utilises a Kronecker product-based feature extraction module and a landmark localisation module to extract crucial landmarks on the body of the prawn. This enables us to perform accurate morphometric analysis and weight estimation from prawn images in a highly efficient manner.

Our proposed approach of using the distance matrix as a feature for prawn weight estimation offers several advantages over traditional methods such as segmentation. It takes into account

the spatial relationships between the landmarks, providing a more accurate estimation of prawn weight. Furthermore, our approach is less sensitive to variations in the prawn's posture or orientation, which can significantly affect the accuracy of traditional methods. Our image processing approach provides non-invasive, accurate prawn measurements, aiding efficient monitoring and decision-making in aquaculture, and promoting sustainability. It is fully automated, saving time and reducing manual labour. Moreover, it is accurate and robust, enabling reliable prawn weight estimation even in challenging environments.

In our proposed deep learning model, we detect landmarks on the prawn body to achieve highly accurate weight estimations. By detecting and analysing these landmarks, other important physical characteristics of the prawn, such as its length, width, and shape, can be extracted. An advantageous aspect of this process is that the morphometric analysis is essential for the weight estimation process.

Automated morphometric analyses using deep-learning-based computer vision and image processing offer several advantages over traditional manual measurements. They are faster, more efficient, and more objective, making them particularly beneficial for large-scale aquaculture operations. By eliminating human bias and error, these automated techniques can provide more reliable and accurate data, which is crucial for effective aquaculture management. The need for such techniques in the aquaculture industry is evident, and their implementation could significantly enhance operational efficiency and productivity.

However, it is important to consider that these findings could be influenced by the specific characteristics of our dataset. For instance, if our dataset has a small distribution in size classes and represents prawns at a specific point along their growth curves (allometric vs isometric), this could impact the effectiveness of PCA. In datasets with a more dispersed distribution of different size classes, PCA might provide additional value by reducing dimensionality while preserving the most important variations in the data. Future studies could explore this aspect further to optimize model performance across different datasets.

One of the main strengths of our work is the unique application of the Kronecker convolution operation for feature extraction. This method captures rich semantic information from various scales of the prawn image, resulting in more accurate landmark detection and weight estimation. The success of our Deep Learning-based morphometric analysis heavily relies on using a well-

curated and annotated dataset. It provides our models with sufficient training data to learn shape information from the images and accurately identify landmark points.

Our approach has demonstrated potential, but it's not without its constraints. The success of our method is heavily reliant on the quality of the input images. Insufficient information for accurate prawn weight estimation may be provided by low-quality images. Moreover, the effectiveness of our Deep Learning architecture is contingent upon the availability of large-scale annotated datasets, which could be a hurdle in certain scenarios. Our experiments were carried out on a single GPU, which might not fully represent the performance profile and could vary with different hardware configurations. The dataset utilized for our experiments was limited in both size and scope. Consequently, to verify the efficacy and applicability of our approach, it's necessary to conduct further studies on more diverse datasets.

In light of these limitations, our future work will focus on several key areas. We aim to explore new strategies to handle low-quality images and develop efficient and automated methods for acquiring and annotating large-scale datasets. We also plan to investigate the relationship between image quality and weight estimation accuracy. Furthermore, we intend to broaden the scope of our approach by applying it to other species of aquatic animals. We will also explore its potential for other applications such as automated disease diagnosis and monitoring. We are confident that our work can make a significant contribution to prawn aquaculture management and production, thereby paving the way for new opportunities in the application of Deep Learning in aquatic animal research and aquaculture engineering.

In conclusion, our use of Deep Learning techniques to extract features from prawn images is a promising area of research that has shown great potential for morphometric analyses. Future work will focus on addressing these limitations and further improving the efficiency and accuracy of our approach.

6 Conclusions

In this paper, we have proposed a novel deep learning approach for automated morphometric analyses and weight estimation of prawns from images. Our approach is efficient, accurate, and has the potential to improve the efficiency and profitability of aquaculture and fisheries operations. We believe that our work can significantly contribute to prawn aquaculture management

and production, opening up new opportunities for deep learning applications in aquatic animal research and aquaculture engineering.

6.1 CO2 Emission Related to Experiments

Experiments were conducted using a private infrastructure, which has a carbon efficiency of 0.432 $kgCO_2eq/kWh$. A cumulative of 500 hours of computation was performed on the hardware of type RTX 2080 Ti (TDP of 250W). Total emissions are estimated to be 54 $kgCO_2eq$ of which 0 percents were directly offset. Estimations were conducted using the MachineLearning Impact calculator presented in [29].

More in detail, in Table 3, we compare our proposed model with a ResNet-18 for landmark detection. We find that our model reduces both training time and carbon emissions by 25%. The ResNet-18 takes about 20 hours and emits 2.16 $kgCO_2eq$, while our model takes about 15 hours and emits 1.62 $kgCO_2eq$. Carbon emissions are a major concern for training large deep-learning models. Therefore, we believe that our method is a small step towards more efficient and eco-friendly models.

ACKNOWLEDGEMENT

This research is supported by the Australian Research Training Program (RTP) Scholarship and Food Agility HDR Top-Up Scholarship. D. Jerry and M. Rahimi Azghadi acknowledge the Australian Research Council through their Industrial Transformation Research Hub program.

ADDITIONAL INFORMATION

Competing interests The authors declare no competing interests.

REFERENCES

[1] N. V. Sang, N. T. Luan, N. V. Hao, T. V. Nhien, N. T. Vu, and N. H. Nguyen, "Genotype by environment interaction for survival and harvest body weight between recirculating tank system and pond culture in Penaeus monodon," *Aquaculture*, vol. 525, p. 735278, 8 2020.

[2] C. E. Boyd, R. P. Davis, and A. A. McNevin, "Perspectives on the mangrove conundrum, land use, and benefits of yield intensification in farmed shrimp production: A review," 2022.

[3] P. Mitteroecker and P. Gunz, "Advances in Geometric morphometrics," *Evolutionary Biology*, vol. 36, no. 2, 2009.

[4] A. Loy, S. Busilacchi, C. Costa, L. Ferlin, and S. Cataudella, "Comparing geometric morphometrics and outline fitting methods to monitor fish shape variability of Diplodus puntazzo (Teleostea: Sparidae)," *Aquacultural Engineering*, vol. 21, no. 4, 2000.

[5] C. Costa, F. Antonucci, C. Boglione, P. Menesatti, M. Vandeputte, and B. Chatain, "Automated sorting for size, sex and skeletal anomalies of cultured seabass using external shape analysis," *Aquacultural Engineering*, vol. 52, pp. 58–64, 2013.

[6] Z. Dellacqua, C. Di Biagio, C. Costa, P. Pousão-Ferreira, L. Ribeiro, M. Barata, P. J. Gavaia, F. Mattei, A. Fabris, M. Izquierdo, and C. Boglione, "Distinguishing the Effects of Water Volumes versus Stocking Densities on the Skeletal Quality during the Pre-Ongrowing Phase of Gilthead Seabream (Sparus aurata)," *Animals*, vol. 13, no. 4, 2023.

[7] A. Loy, M. Bertelletti, C. Costa, L. Ferlin, and S. Cataudella, "Shape changes and growth trajectories in the early stages of three species of the genus Diplodus (Perciformes, Sparidae)," *Journal of Morphology*, vol. 250, no. 1, 2001.

[8] M. Jahanbakht, W. Xiang, L. Hanzo, and M. R. Azghadi, "Internet of Underwater Things and Big Marine Data Analytics - A Comprehensive Survey," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 2, pp. 904–956, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9328873/

[9] D. A. Konovalov, A. Saleh, D. B. Efremova, J. A. Domingos, and D. R. Jerry, "Automatic Weight Estimation of Harvested Fish from Images," in *2019 Digital Image Computing: Techniques and Applications, DICTA 2019*. Institute of Electrical and Electronics Engineers Inc., 12 2019.

[10] A. Saleh, M. Sheaves, and M. Rahimi Azghadi, "Computer vision and deep learning for fish classification in underwater habitats: A survey," *Fish and Fisheries*, vol. 23, no. 4, pp. 977–999, 7 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/faf.12666

[11] A. Setiawan, H. Hadiyanto, and C. E. Widodo, "Shrimp Body Weight Estimation in Aquaculture Ponds Using Morphometric Features Based on Underwater Image Analysis

and Machine Learning Approach," *Revue d'Intelligence Artificielle*, vol. 36, no. 6, pp. 905–912, 12 2022.

[12] T. T. E. Vo, H. Ko, J. H. Huh, and Y. Kim, "Overview of Smart Aquaculture System: Focusing on Applications of Machine Learning and Computer Vision," *Electronics 2021, Vol. 10, Page 2882*, vol. 10, no. 22, p. 2882, 11 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/22/2882/htmhttps://www.mdpi.com/2079-9292/10/22/2882

[13] I. H. Laradji, A. Saleh, P. Rodriguez, D. Nowrouzezahrai, M. R. Azghadi, and D. Vazquez, "Weakly supervised underwater fish segmentation using affinity LCFCN." *Scientific reports*, vol. 11, no. 1, p. 17379, 12 2021. [Online]. Available: https://www.nature.com/articles/s41598-021-96610-2http://www.ncbi.nlm.nih.gov/pubmed/34462458http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC8405733

[14] A. Saleh, I. H. Laradji, D. A. Konovalov, M. Bradley, D. Vazquez, and M. Sheaves, "A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis," *Scientific Reports*, vol. 10, no. 1, p. 14671, 12 2020. [Online]. Available: https://www.nature.com/articles/s41598-020-71639-x

[15] J. V. Devi, S. G. Deo, and R. Khandeparkar, "Kronecker Product," in *Linear Algebra to Differential Equations*, 2021.

[16] D. A. Konovalov, A. Saleh, D. B. Efremova, J. A. Domingos, and D. R. Jerry, "Automatic weight estimation of harvested fish from images," in *Digital Image Computing: Techniques and Applications (DICTA)*, 2019, pp. 1–7.

[17] N. Bravata, D. Kelly, J. Eickholt, J. Bryan, S. Miehls, and D. Zielinski, "Applications of deep convolutional neural networks to predict length, circumference, and weight from mostly dewatered images of fish," *Ecology and Evolution*, 2020.

[18] D. Hung, N. H. Nguyen, D. A. Hurwood, and P. B. Mather, "Quantitative genetic parameters for body traits at different ages in a cultured stock of giant freshwater prawn (Macrobrachium rosenbergii) selected for fast growth," *Marine and Freshwater Research*, vol. 65, no. 3, p. 198, 2014. [Online]. Available: http://www.publish.csiro.au/?paper=MF13111

[19] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Lecture Notes in Computer*

Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014.

[20] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A Comprehensive Survey of Image Augmentation Techniques for Deep Learning," *Pattern Recognition*, vol. 137, p. 109347, 5 2023.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015, pp. 234–241.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[23] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6848–6856.

[24] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation," *CoRR*, vol. abs/1801.0, 2018.

[25] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and ¡0.5MB model size," *Computer Vision and Pattern Recognition*, 2 2016. [Online]. Available: http://arxiv.org/abs/1602.07360

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019.

[28] M. M. Hasan, P. C. Thomson, H. W. Raadsma, and M. S. Khatkar, "Genetic analysis of digital image derived morphometric traits of black tiger shrimp (Penaeus monodon) by incorporating G E investigations," *Frontiers in Genetics*, vol. 13, 10 2022. [Online].

Available: https://pubmed.ncbi.nlm.nih.gov/36338959/

[29] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, "Quantifying the Carbon Emissions of Machine Learning," 10 2019. [Online]. Available: /green-ai/publications/2019-11-lacoste-quantifying.html

Alzayat Saleh: Conceptualisation, Data Curation, Data Analysis, Software Development, DL Algorithm Design, Visualization, Writing original draft. **Md Mehedi Hasan:** Data Curation, Data Analysis. **Herman W Raadsma**: Data Curation, Data Analysis, Reviewing/editing the draft. **Mehar S Khatkar**: Data Curation, Data Analysis, Reviewing/editing the draft. **Dean Jerry**: Conceptualisation, PhD Supervision, Reviewing/editing the draft. **Mostafa Rahimi Azghadi**: Conceptualisation, Data Curation, Data Analysis, PhD Supervision, Reviewing /editing the draft.

Declaration of interests

☐ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Alzayat Saleh, Dean Jerry, Mostafa Rahimi Azghadi has patent pending to James Cook University.

Highlights

- Novel Deep Learning approach automates weight estimation and morphometric analysis in aquaculture.
- Approach applied to black tiger prawn (Penaeus monodon) as a model crustacean.
- Feature extraction and landmark localization modules used to predict key morphological points.
- Weight estimated using a regression module based on extracted landmarks.
- Approach outperforms existing methods in terms of accuracy, robustness, and efficiency.