UiA University of Agder

Master

Identification of Irregularities in Salmon Fish in Aquaculture by Using Key Point Detection

This project aims to deliver an AI based solution to find out irregularities and deformities in farmed salmon fish by using Keypoint detection

YERONIS ASSEFA HUBENA
MOHAMED ALI ABDULLAHI

SUPERVISOR
Aditya Gupta

University of Agder, 2023

Faculty of Engineering and Science

Department of Engineering and Sciences

Obligatorisk gruppeerklæring

Den enkelte student er selv ansvarlig for å sette seg inn i hva som er lovlige hjelpemidler, retningslinjer for bruk av disse og regler om kildebruk. Erklæringen skal bevisstgjøre studentene på deres ansvar og hvilke konsekvenser fusk kan medføre. Manglende erklæring fritar ikke studentene fra sitt ansvar.

1.	Vi erklærer herved at vår besvarelse er vårt eget arbeid, og at vi ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.	Ja / Nei
2.	**Vi erklærer videre at denne besvarelsen:** • Ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands. • Ikke refererer til andres arbeid uten at det er oppgitt. • Ikke refererer til eget tidligere arbeid uten at det er oppgitt. • Har alle referansene oppgitt i litteraturlisten. • Ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.	Ja / Nei
3.	Vi er kjent med at brudd på ovennevnte er å betrakte som fusk og kan medføre annullering av eksamen og utestengelse fra universiteter og høgskoler i Norge, jf. Universitets- og høgskoleloven §§4-7 og 4-8 og Forskrift om eksamen §§ 31.	Ja / Nei
4.	Vi er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert.	Ja / Nei
5.	Vi er kjent med at Universitetet i Agder vil behandle alle saker hvor det forligger mistanke om fusk etter høgskolens retningslinjer for behandling av saker om fusk.	Ja / Nei
6.	Vi har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider.	Ja / Nei
7.	Vi har i flertall blitt enige om at innsatsen innad i gruppen er merkbart forskjellig og ønsker dermed å vurderes individuelt. Ordinært vurderes alle deltakere i prosjektet samlet.	Ja / Nei

Publiseringsavtale

Fullmakt til elektronisk publisering av oppgaven Forfatter(ne) har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten (Åndsverkloven. §2).
Oppgaver som er unntatt offentlighet eller taushetsbelagt/konfidensiell vil ikke bli publisert.

Vi gir herved Universitetet i Agder en vederlagsfri rett til å gjøre oppgaven tilgjengelig for elektronisk publisering:	Ja / Nei
Er oppgaven båndlagt (konfidensiell)?	Ja / Nei
Er oppgaven unntatt offentlighet?	Ja / Nei

Acknowledgements

We would like to express our gratitude to our team for their dedication and hard work in order to complete this project successfully. We would also want to express our gratitude to Aditya Gupta, our project supervisor, for his guidance and advice during the project's development.

Abstract

This thesis focuses on finding solutions for challenges that arise within aquaculture. These challenges include wound and lice infestations, diseases, and deformities in the fish's physical structure. Seafood plays a great role in the world's economy, as several countries depend on seafood for consumption. Hence, aquaculture is vital to keeping sea animals healthy and secure for dietary consumption. In order To address this issue and achieve precise measurement of individual fish, this study proposed automatic salmon fish deformity detection by using keypoint detection. We have utilized datasets that contain the key points of the fish and the bounding boxes. R-CNN and the YOLO model were used in this project for keypoint detection purposes. The learning process of the models was quite efficient, as both R-CNN and YOLO showed good results in detecting key points on the fish. The YOLO model has excelled in detecting keypoints with a mean average precision of 98.3% compared to the R-CNN model with a mean average precision of 96%. However, R-CNN has shown better keypoint alignment, meaning a very precise keypoint calibration compared to the YOLO model.

These key point detection facilitate precise geometrical analyses, such as assessing triangles, slopes, and distance pairs for further investigation to identify any irregularities in the fish. The method that was proposed for this study was to categorize the irregularities into three different categories, such as jaw, size (thin or thick) and spine deformity. Merging the key point detection model with geometrical analysis confirms their ability to identify deformities in the salmon fish body. The result indicates the proposed methods' ability to accurately detect key points, offering superior accuracy compared to other methods.

Keywords: Aquaculture, Deep Learning, CNN, Keypoint Detection, R-CNN, YOLO, Geomtric Analysis, Distances, Triangle, Slope

Contents

Acknowledgements **iii**

Abstract **iv**

List of Figures **viii**

1 Introduction **1**

2 Background **3**

2.1 AI applications in aquaculture 3

2.2 Convolutional Neural Network 3

2.3 Object Detection 5

2.3.1 R-CNN 5

2.3.2 Faster R-CNN 6

2.3.3 Mask R-CNN 6

2.3.4 YOLO 7

2.4 Augmentation 7

3 State of the Art **9**

3.1 Fish Key points Detection for Ecology Monitoring Based on Underwater Visual Intelligence 9

4 Method **11**

4.1 Model 11

4.1.1 Keypoint RCNN 11

4.2 Data pre-processing 12

4.2.1 Data Augmentation 13

4.2.2 YOLO 13

4.3 Performance Metrics 14

4.4 Geometrical Analyse 15

4.4.1 Approach One 15
4.4.2 Approach Two 17

5 Result and Discussion **26**
5.1 Literature Review: Comparison of results with Other researches 27
5.2 Geometrical Analyses 28
5.3 Case 1: Size-Related Irregularities Results 29
5.3.1 Case 2: Jaw Deformity Result 32
5.3.2 Case 3: Spine Deformity Result 34
5.4 Challenges 37

6 Conclusions **39**
6.1 Conclusion 39
6.2 Future Work 40

Bibliography **41**

List of Figures

2.1 R-CNN 6
2.2 Mask R-CNN 7

4.1 Customized Keypoint R-CNN Architecture: Incorporates 20key points, a modification from the standard 17, to align with our use-case demands. . . . 12
4.2 pre-processing and workflow 13
4.3 Data Augmentation 13
4.4 Fish with triangles 14
4.5 Precision 14
4.6 Recall 15
4.7 Fish with key points alphebta 16
4.8 15 unuque triangles was formed Fish 16
4.9 42 Distance pairs was formed using across the entire fish body 17
4.10 Categorizing irregularities into three categories.s 18
4.11 Size-Relate annotation 19
4.12 Jaw annotation with 2 triangles and their slopes 21
4.13 Jaw distance annotation on the fish jaw 22
4.14 Distances Pairs and their connection each other 23
4.15 Spine annotation 24
4.16 Spine distance annotation on the fish spine 25
4.17 Distances Pairs for spine and their connection each other 25

5.3 Using Approach 1 Detected Deformity 28
5.4 Using Approach 1 Detected No Deformity 28
5.5 workflow 29
5.6 Abnormal Thickness and Size Deformities 30
5.7 This fish has a type of thin deformity. 31
5.8 Thin deformity 32
5.9 Detected Jaw Deformity 34

5.10 Detected Jaw Deformity 34
5.11 Result that was obtained using distance pairs 34
5.12 Detected Spine Deformity 36
5.13 Result that obtained from detected key points with spine deformity 36
5.14 Detected Spine Deformity 36
5.15 Detected Spine Deformity 37
5.16 Result obtained from detected key points with spine deformity 37

Chapter 1

Introduction

Aquaculture is one of the developing vital solutions for the highly growing and in-demand food market, which is the seafood market, by reducing pressure from the overutilization of fish stocks[41, 26]. In order to see the integration between these new cutting-edge technologies and aquaculture, this research dives into the fish farming world, where artificial intelligence, deep learning, CNN, keypoint detection, object detection, and so many other advanced technologies come together to revolutionize the seafood market[41]. This project aims to navigate the relationships between the different AI-related technologies and how they can be used to control any irregularity or deformity on a fish's body in a fish farm. Another objective of this research is to evaluate to what extent AI-supported solutions can be practical and effective in real-world fish farm settings. It also delves into the implications of utilizing AI technologies in fish farms for the economy's resource efficiency and sustainability[11]. There is a very high demand for the seafood market in the world considering its nutritional importance, and aquaculture plays a great role in the safety and security of seafood[42]. However, aquaculture has its own challenges, such as environmental sustainability, disease management, and precision farming. Hence, AI-related technologies play a great role as they have the ability to process a large amount of data, which helps in making precise decisions.

Deep-learning methods based on convolutional neural networks have improved visual recognition. In recent years, many object detection methods have been proposed, such as Faster R-CNN, YOLO, and IoUNet. In addition, Keypoint detection methods such as Stack Hourglass and R-CNN keypoint detection have also been developed[15]

In our case, AI can propose solutions for challenges such as fish irregularities, which are vital for sustainable and efficient aquaculture. By using AI, fish farms can advance environmental sustainability, improve resource usage, and enhance the general quality and health of fish in aquaculture, which is a very essential mechanism for worldwide food security and economic conservation[40, 26].

Hence, this Master's thesis will contribute to the ongoing research on aquaculture by focusing on finding solutions for the above challenges and methods on how we can identify fish with irregularities and deformities by using keypoint detection and geometrical analysis to create a very efficient and ecological fish farm. To achieve this, we will be utilizing methods such as the YOLO algorithm and RCNN keypoint detection model, as well as the 469 annotated dataset that contains images of salmon fish, keypoints for each image of the salmon fish and bounding boxes, to train and test these models.[41]

Chapter 2

Background

2.1 AI applications in aquaculture

AI has become a very important field in aquaculture. Many companies, including startups, are mainly focusing in AI-based application development. AI plays a great role in monitoring fish farms by using technologies such as object detection, image classification and segmentation, and speech and visual recognition[41]. This enables us to recognize and make decisions without any human supervision. In a research paper titled "Accurate Wound and Lice Detection in Atlantic Salmon Fish Using a Convolutional Neural Network", written by Aditya Gupta, Even Bringsdal, Kristian Muri Knausgård and Morten Goodwin, they addressed challenges such as sea lice infestations, wounds in fish, irregular fish maturity, diseases, and other irregularities that danger fish mortality[16]. They have stated that current solutions, such as clean fish measurement and parasite drugs, are used to reduce sea lice; however, it is difficult to eliminate them completely. They have proposed a CNN model that achieves a high accuracy of 96.7% [16] in detecting wounds and lice. This research mainly uses classification to find a solution, while Our research takes a different approach to finding irregularities by using key point detection. The research further explains the importance of cost-efficient and highly effective automated methods for wound and lice detection in order to protect fish in aquaculture. There are several types of research that support the advancement of AI in aquaculture to protect fish farms as the coastal population highly relies on sea food as a food source, which leads to high demand and growth of fish farms[41, 43].

2.2 Convolutional Neural Network

Neural networks are known for forming the base for deep learning as they are the subset of ML algorithms[8]. They are made up of layers such as the input layer, a single or several hidden layers, and an output layer with interconnected nodes. And these nodes, known as

neurons, are connected with each other, and each has a weight and threshold. Data is transmitted when the output of a neuron is above the specified threshold, then it gets activated and the data will be passed to the next layer. However, if it is below the threshold, no data will be transmitted. The two most prominent examples of a neural network are the convolutional Neural Network (CNN), which is mostly used for image classification and computer vision, and the Recurrent Neural network (RNN). RNN is mostly used for speech recognition as well as language processing[10]. In this section, we will discuss the convolutional neural network, CNN. The traditional method for image recognition demanded a labor intensive extraction process. However, CNN played a great role in automating the feature extraction from an image[33]. The three components CNN consists of include; the Convolutional layer, the pooling layer, and the Fully connected (FC) layer[44]. The convolutional layer is the main layer of CNN, where most of the processes take place. It is composed of a few components such as the input data, which is a 3D matrix that represents colored images with height, width, and depth; a feature detector, such as a filter or a kernel, which performs a process called convolution, making a 2D array of weights that slides across the receiving fields, and for each position it is in, it performs a dot product to check if the feature is present in the image; and a feature map, which is a result produced by the convolution process of the feature detector and is an output of the dot product from the input and the feature detector. [7]. The pooling layer is another layer of the neural network. It reduces dimensionality or downsampling to minimize the number of parameters in the input. This layer is a similar process as the convolutional layer when it comes to sliding a filter across the information; however, the filter doesn't have any weight. There are two pooling methods that are commonly used: max pooling, which selects the pixel with the highest value as the filter slides across the input and sends it to the output array; and average pooling, which calculates the average value in the field as the filter moves across the input to send it to the output array. In general, pooling helps minimize complexity, increase computational time, and prevent overfitting by making parameter reductions[44]. In layers that are partially connected, the value of the pixel in the input image is not directly connected to the output layer. However, in the Fully Connected layer, every node in the output layer is connected to a node in the former layer. The FC layer is mainly responsible for classification by using the extracted features from previous layers and filters. It also uses the softmax activation function to undergo the input classification process, resulting in a probability of 0 - 1 [37][44][38]. CNN is a very important mechanism in deep learning and AI in general, as it mainly focuses on image processing and its strong ability to identify unique patterns by utilizing feature extraction[8][44] [34]

2.3 Object Detection

Object detection is fundamental for computer vision tasks which involve identifying and classifying objects in an image or video [6]. Object detection is useful for recognizing and locating objects within the visual data. Object detection utilizes bounding boxes as visual markers for the identified object [12]. In this section, the most commonly used object detection algorithms are discussed.

2.3.1 R-CNN

R-CNN (Region-based Convolutional Neural Network) is an architecture that is designed for object detection. The architecture of R-CNN unfolds in three steps [13, 35] as shown in Figure 2.1. To undergo RCNN, the first stage is the Region Proposal Generation. In this stage, the input image is analyzed to pinpoint potential areas where the object might be located. This is achieved using an algorithm known as the Selective Search, which generates 2,000 region proposals for each image[27]. The next step is the CNN application stage, where each regional proposal is fed into CNN. The CNN's role is to extract features from the proposal, lay the groundwork for accurate classification and bounding box adjustment, and serve as input for the subsequent stage. In the final 'Processing the Outputs' stage, the features extracted from each proposal are then fed into two different mechanisms tailored for each region. One of these mechanisms uses the Support Vector Machine (SVM) to classify the region of the object in an image. Simultaneously, the other mechanism uses a linear regressor to refine and adjust the bounding box around the detected object for a better fit. However, it is worth mentioning that the R-CNN model has its own limitations and challenges. One of the most significant limitations is that it requires a considerable amount of training time. This is because it has to process almost 2000 region proposals each time. Such demands can be impractical for real-time scenarios where processing a single image could take almost 47 seconds[30]. Furthermore, the intricate nature of the R-CNN framework demands the separate training of three models: the CNN for feature extraction, the SVM for object classification, and a regressor to optimize bounding boxes. Due to this limitation, the R-CNN model required enhancements, leading to the development of Fast R-CNN.[13]

The main reason that the Fast R-CNN is faster than the R-CNN is that it does not need to feed 2000 region proposals into the convolutional neural network for each image. Instead, the convolutional operation is done for each image in the form of a single-forward operation. It is only fed once in order to generate the feature map. The training of CNN, classifier, and bound box regressor of fast R-CNN is done simultaneously in a single model.[31]

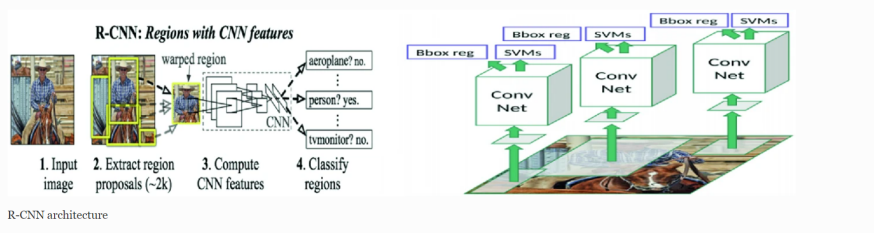


Figure 2.1: R-CNN

Despite all these advantages, the region proposal is the main challenge that Fast R-CNN is facing. The selective search is used for the proposal, and it slows the process and becomes a bottleneck of overall performance.[13]

2.3.2 Faster R-CNN

Faster R-CNN is an enhancement over previous models, R-CNN and Fast R-CNN. One of its major improvements is the replacement of the selective search method with the Region Proposal Network (RPN). [30, 31]

In the Faster R-CNN workflow, the input image is fed to the CNN which provides the convolutional feature map. RPN is used instead of the Selective search algorithm on the feature map to identify the region proposal. The convolutional feature maps of the CNN undergo a transformation such as downscaling to dimension, for instance, 256-d by a 3x3 [13, 31] sliding window. The RPN is responsible for creating a variety of region-based anchor boxes, each maintaining a fixed ratio of k for each location of the sliding window [13, 31, 9]. These boxes can vary in shape; some could be large, wide, tall, etc. In order to determine whether these boxes contain an object, the Intersection-over-Union (IoU) is used with a threshold of more than 0.7 which is a strong indication of the object's presence; otherwise, if the Intersection-over-Union (IoU) is less than 0.3, it indicates there is no object. This region is not processed immediately. First, they navigate through the RoI pooling layer, followed by fully connected layers. At last, the output of this is fed to the softmax classification layer and bounding box regressor [9, 22]

2.3.3 Mask R-CNN

The previous model was R-CNN. The R-CNN could classify the object in the image, but it could not show exactly which pixels belonged to the object. Hence, here is where the Mask R-CNN plays its role. The Mask R-CNN is advanced by the Faster R-CNN, which is capable of showing exactly which specific pixel makes up an object in an image [2]. Mask R-CNN has two processes. The first one is to identify the drawing that contains the object detection part, and it uses Faster R-CNN for this. The second is that it colors inside the

lines, which are pixel-level details, which is called semantic segmentation. Mask R-CNN uses a method called RoInAlign. It uses the ROI-Align layer in place of the traditional ROI-pooling. This substitution improves accuracy, especially in segmentation tasks. The replacement is crucial because the ROI-pooling layer often discards vital information during the quantization process. In contrast, the ROI-Align method in Mask R-CNN leverages bilinear interpolation to translate values from a variable-sized feature map to a fixed-size one and preserves exact spatial locations.[9, 2]. Moreover, Mask R-CNN can generalize tasks outside its primary training, such as estimating human poses and person Key point detection [2] .

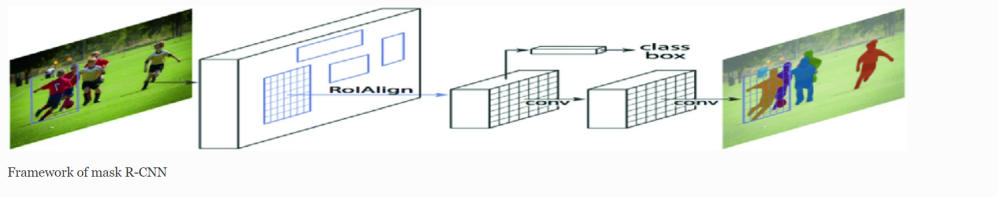


Figure 2.2: Mask R-CNN

2.3.4 YOLO

YOLO or You Only Look Once, is one of the most advanced and important algorithms for object detection which uses a unique method by combining and solving tasks such as the prediction of bounding boxes and class probabilities at the same time by using an end-to-end neural network[47]. The traditional method of object detection algorithms takes separate steps for these tasks, which makes YOLO more effective and efficient. In the YOLO architecture, the image is divided into SxS grid, where each and every grid cell has its own role of detecting the object to which the center of the object falls[46]. In order to indicate the possibility of an object being present, each cell predicts the bounding box and confidence scores, which also shows the accuracy of the predicted box[47]. YOLO improves recall scores by using a specialization technique to allocate a bounding box predictor for each object. YOLO applies a technique called Non-maximum Suppression (NMS), to help eliminate redundancy of bounding boxes or incorrect bounding boxes and as a result, it outputs a single bounding box for each object[46] [48].

2.4 Augmentation

Data Augmentation is another part of the Convolutional neural network, which applies several transformations such as rotation, resizing, flipping, padding, and cropping, in order to artificially generate new training data derived from an existing one[45, 5]. This method

helps machine learning models improve performance and robustness in tasks such as image recognition. The main purpose of training machine learning models is to recognize objects under several conditions, such as zoom level, rotation, different lighting, and noise. Data augmentation is useful for solving data overfitting and security issues, which is why models are more robust and perform well. CNN needs a lot of data training for effective and efficient learning. However, data augmentation can expose the model to a wide range of variations of real-life scenarios. Data augmentation's role is vital in machine learning, as it uses a realistic variation of the original data to expand the training data set artificially. [45].

Chapter 3

State of the Art

3.1 Fish Key points Detection for Ecology Monitoring Based on Underwater Visual Intelligence

The research paper titled "Fish Key points Detection for Ecology Monitoring Based on Underwater Visual Intelligence" written by Feiyang Suo, Kangwei Huang, Gui Ling, Yanjun Li, and Ji Xiang, suggests a fisher monitoring system for the ecology of fish in cultivation pools[39]. It uses stereo Key point detection, as well as Deep learning to find an estimation of the posture and length of a fish. The system used for monitoring fish ecology uses a deep neural network and underwater visual intelligence to undergo precise monitoring. The traditional way of fish ecology monitoring involves manually measuring the fish by catching a fish and making it as a sample to decide their posture and size. This approach is a labor intensive, as well as intrusive method, hence, this paper addresses the necessity of having a non-invasive monitoring system in aquaculture and highly depends on technologies such as computer vision and deep learning in monitoring real-time fish population.[39] In order to undergo this, the researchers have used several methods. The first step was fish detection. In this step, the system undergoes fish detection from the images taken by binocular cameras by applying a deep learning model such as Faster R-CNN. Then this leads to generating bounding boxes around each fish. The next step is to match Stereo bounding boxes to identify similar fish in the left image and right image by using similarity measures. Then it expands the bounding box to include more important context, which helps with detecting the correct key points on the image of the fish. Key point detection is the next step and this is done by employing stacked hourglass architecture which detects key points such as the eye, head, gills, dorsal fin, etc[14]. And these help find information about the posture of the fish. Once this is done, the next step is to use the detected key points to estimate the posture of the fish by fitting a curve in 3-dimensional space, and the curve's length is used

to estimate the length of the fish. As a result, the system is first trained and tested based on an underwater database for fish. When it comes to fish detection, it was proven that Faster R-CNN attains a higher accuracy mAP (mean average precision)[39]. For key point detection, it is tested using Object Key point Similarity, OKS, and for every stage there was a significant improvement in accuracy. They have found the estimated length of the fish to be quite accurate, with an average error rate of 5.58 % even if it's a curved or slanted fish. In conclusion, the system that is used for fishery ecology monitoring uses different methods, such as Key point detection, fish detection, and fitting curves in 3D space, in order to accurately estimate the posture and length of the fish. The system successfully overcomes limitations such as illumination problems underwater and pixel-wise matching problems in stereo images. It was proven that this system shows better accuracy and efficiency in comparison to the traditional manual methods, which makes it quite important for real-time aquaculture ecology monitoring systems. [39] [14]

Chapter 4

Method

4.1 Model

4.1.1 Keypoint RCNN

Mask RCNN is a Convolutional neural network developed to process instance segmentation, and it is capable of identifying the object at the pixel level. In particular, it encompasses functionalities for both object detection and instance segmentation, providing an adaptable approach to image analysis[25]. Mask RCNN experiences two phases of training. Initially, it generates a region proposal where the object might be present in an input image. The second phase is to use the proposed region to predict the object class and construct a bounding box around the object detected. Feature extraction of a raw image was performed[25].

Using Mask R-CNN, enhanced with ResNet50 and the Feature Pyramid Network (FPN), as our backbone network [36], ensures efficient feature extraction from images. ResNet50, a form of residual network, employs skip connections in forward and backward propagation to mitigate vanishing and exploding gradient issues, stabilizing and optimizing the model training process [3].

In contrast to Mask R-CNN, Keypoint R-CNN encodeskey points of objects rather than their entire mask [20], showcasing a contrasting approach to object representation. This Figure 4.1 illustrates the architecture of thekey point R-CNN. Box head takes the feature map provided by the backbone network and uses it to perform two tasks. Classifying objects and predicting their bounding boxes. Object classification involves identifying what type of object is present while bounding box regression involves predicting the coordination of a box that encloses an object[20]. Enhancing the model to accurately predict key points on detected objects involves integrating the key point detection head with the bounding head and employing comprehensive network training. Combining this method with a key point head can be used to estimate the human pose as shown in a previous study[4]. As

we have 20 key points in the fish, we adopt the following architecture in this study, which is Mask R-CNN based on ResNet50-FPN with a key point detection module. This allows us to identify the boundary box and 20 key points within each box. With 400 training and 69 test datasets, we used a batch size of 3, incorporating randomness by shuffling, while the test batch size was set to 1 without shuffling for efficient evaluation. During optimization, a stochastic gradient optimizer was utilized with a learning rate of 0.01, a momentum of 0.9 and a weight decay of 0.005, guided through 100 epochs by a learning rate scheduler with a step size of 5 and a gamma of 0.3. The model weights were then saved for evaluation purposes.

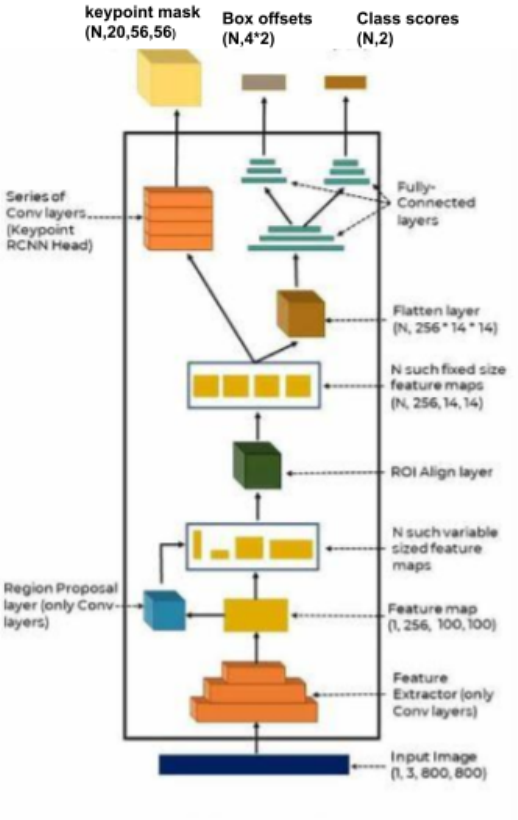


Figure 4.1: Customized Keypoint R-CNN Architecture: Incorporates 20key points, a modification from the standard 17, to align with our use-case demands.

4.2 Data pre-processing

Utilizing COCO for annotation, each image incorporated annotations of key points, bounding boxes, and binary visibility state, with 2 indicating visible key points and 0 for invisible key points. Each image consists of 20 key points, each presented in the form of (x, y) coordinate and visibility, summarized in a 3x20 matrix. Annotations of images are stored in a JSON file, where they are mapped to their corresponding images. Data was subsequently split into training and test sets, where 69 were used for testing the model, while 400 of the dataset was

used for training purposes. Figure 4.2 illustrates the workflow of the pre-processing stage.

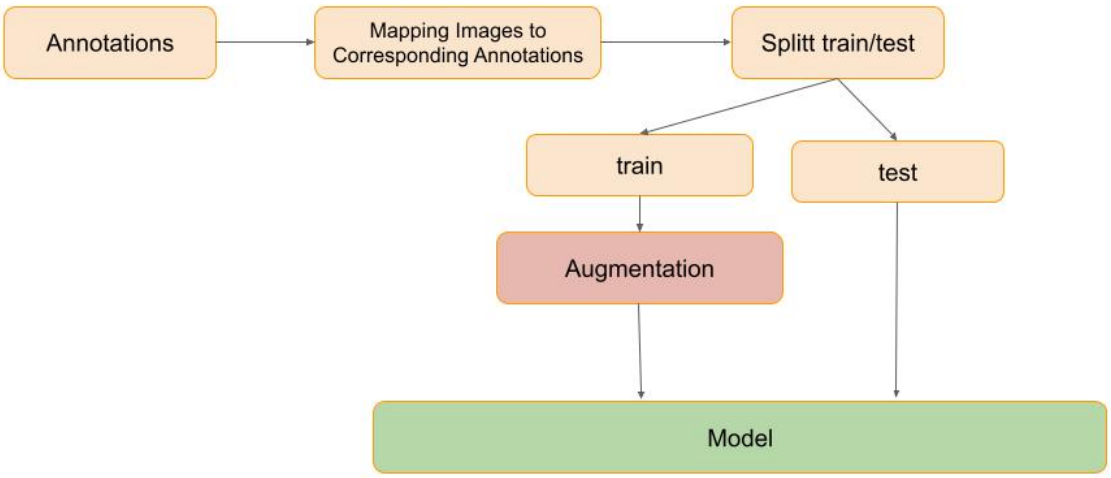


Figure 4.2: pre-processing and workflow

4.2.1 Data Augmentation

In this project, data augmentation was introduced to improve the robustness and generalization of the model. Various augmentative transformations were applied to the training dataset without deviating from realistic image conditions.

Different transformations were implemented, such as rotation, brightness and contrast blurring, and color jittering. Each transformation was carefully parameterized to ensure the augmented data remained realistic and representative of potentially unseen data.

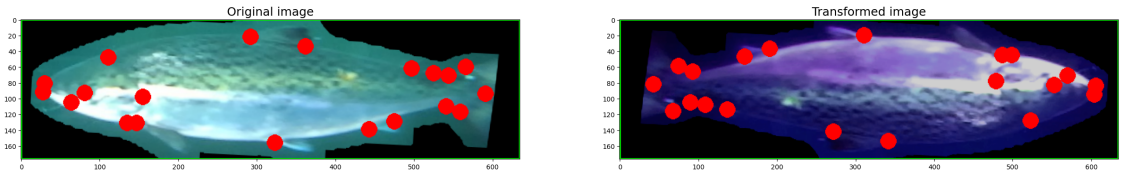


Figure 4.3: Data Augmentation

4.2.2 YOLO

YOLO (You Only Look Once) utilizes object detection and image segmentation. The latest version, YOLOv8, is an improvement of the previous version, which was YOLOv5. YOLOv8 is built on cutting-edge advancement in deep learning and computer vision, offering high performance in terms of speed and accuracy[18]. It is suitable for different applications due

to its uniqueness, simple usability, and adapts to different hardware platforms, [19]. YOLO has been actively used in pose estimation by introducing the key point of the body's joints. Such landmark is a distinctive feature and is usually presented as a set of 2D[x,y] or 3D[x,y, visibility] coordinates. Pose estimation can be helpful in scenarios where it is essential to find specific parts of an object in an image and their location relative to each other [49]. The YOLOv8 was trained in human pose estimation, which had 17 key points. However, we had been given images with 20 different key points for this project, and it was necessary to adapt and customize to fulfill our goal.

4.3 Performance Metrics

To assess the accuracy of the model's predicted bounding box compared to the actual ground truth bounding box, metrics are essential for evaluating the model's performance. Intersection Over Union (IoU) is utilized for this purpose, calculating the ratio of the overlap between the predicted and ground-truth boxes to their collective covered area[21]. Thus, IoU provides a measure of alignment between the predicted bounding box and the actual box [24].

Figure 4.4: Fish with triangles

By using IoU various types of threshold, we can identify a detection as being correct or incorrect. By assigning a certain threshold and comparing it with IoU, it can be classified that the detected bounding box is considered correct if the given IoU is higher than the given threshold. However, it can be considered incorrect in the case of IOU being less than the given threshold[29]. Precision P and recall R are mostly used for object detection[17]. Precision is the ability of the model to identify the relevant objects correctly. It tells us what proportion was correctly identified. When the model does not produce a false positive, the precision becomes 1[24] as illustrated in the formula below:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{TP}}{\text{all detections}}$$

Figure 4.5: Precision

Recall is a measurement of the ability of the model to precisely find all ground-truth bounding boxes. It gives the proportion of actual positive that was identified correctly. When the model does not produce a false negative, it has a recall of 1.[24]. The formula for recall is as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\text{all ground truths}}$$

Figure 4.6: Recall

Using AP(Average Precision), the model can be evaluated, and gives the area of under curve (AUC), and PRECISION Recall Curve. Having a larger area means a higher AP, which also means a better machine-learning model. The mAP(mean average precision) is an average of the AP values, which is a further average of the APs for all classes[1].

4.4 Geometrical Analyse

4.4.1 Approach One

The main objective of approach one was to discern deformation in fish skeletons using Geometric analysis. Geometric analysis includes triangles and distances that could provide insight into the fish's structural integrity. As illustrated in **Figure** 4.8, 15 distinctive triangles are formed by connecting three unique key points on the fish body. Additionally, it was constructed with 42 unique edges connecting different pairs of key points and outlined in this **Figure** 4.9. Each triangle and edge offers a unique geometric perspective and contributes to the comprehensive structural analysis. This methodology facilitates the identification of irregularities in the fish by providing a detailed examination of its structure, thus presenting a robust mechanism for analyzing any potential structural abnormalities. Due to the number of parameters, we did not process this approach, and instead, we selected approach Two, which we will explain in the section below4.4.2.

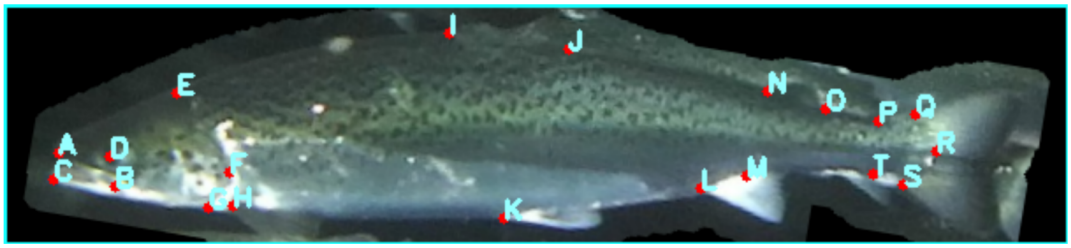


Figure 4.7: Fish with key points alphebta

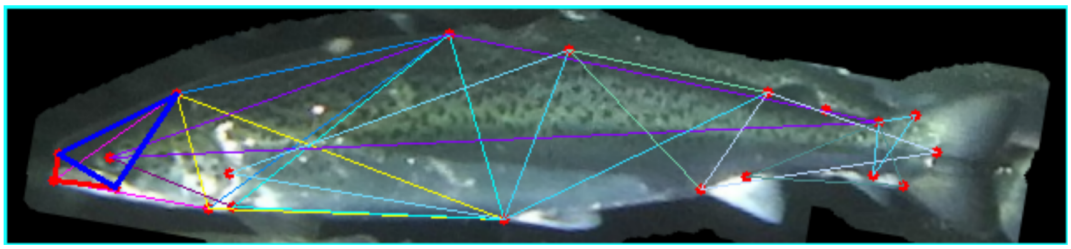


Figure 4.8: 15 unuque triangles was formed Fish

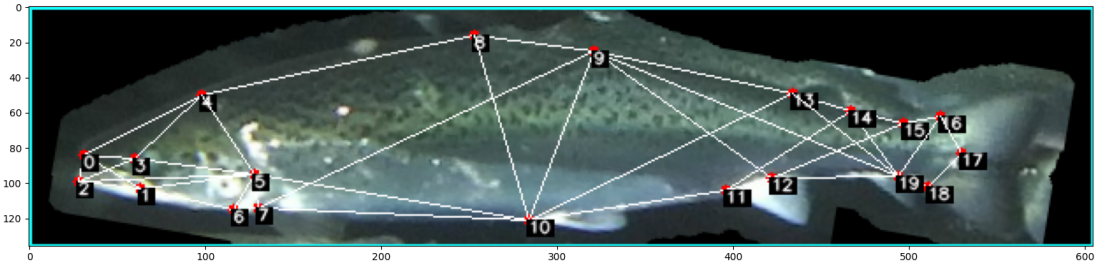


Figure 4.9: 42 Distance pairs was formed using across the entire fish body

4.4.2 Approach Two

As discussed in the previous approach, the procedure seems to be an extensive examination to detect irregularity using 15 triangles and 42 edges. However, the second approach divides the deformity into three categories such as jaw, spine, and Size-Related deformities. Each of these categories was assigned a specific geometric analysis that had to be conducted in order to gain information that is related to that specific category. This method has fewer parameters and provides a more targeted perspective to detect irregularities. These categories outline cases, and each case presents distinct details that guide our analysis.

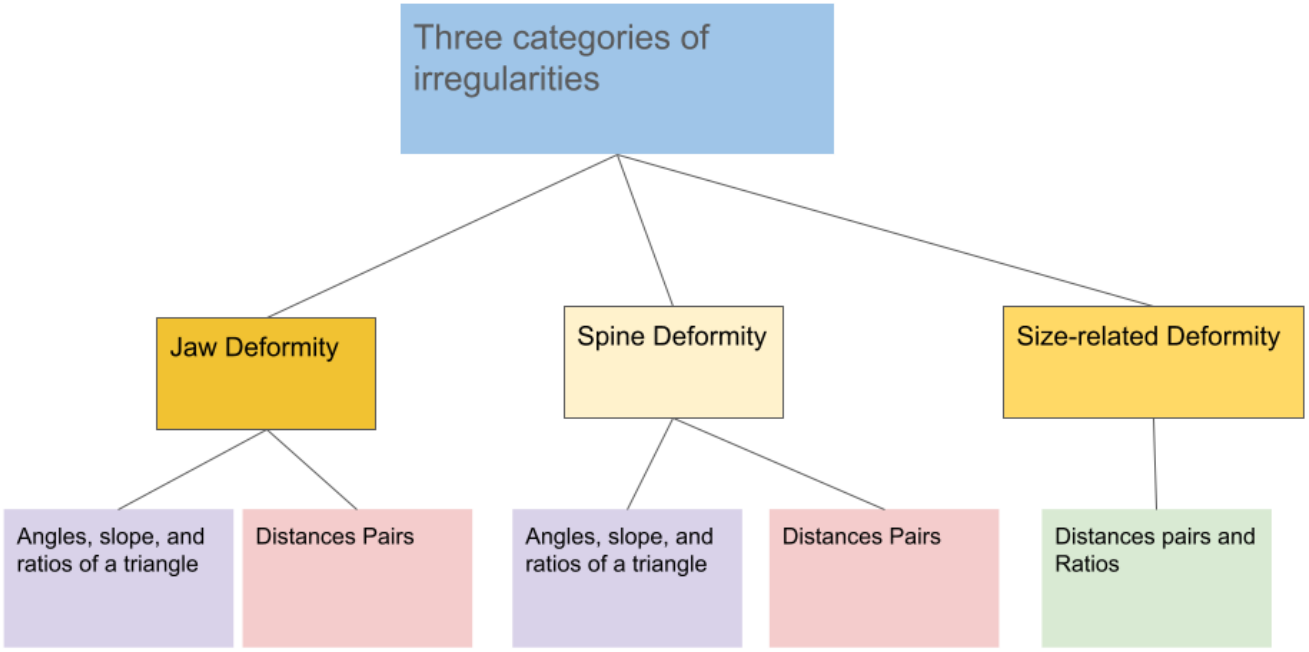


Figure 4.10: Categorizing irregularities into three categories.s

As illustrated in Figure 4.10, each category has distinct measurements associated with its specific characteristics. Certain categories have two tables, as shown in the figure above. For instance, the jaw category includes two tables; the first table includes angles, slopes, and ratios of triangles, while the second contains distance pairs. This dual-table approach proves beneficial as one table might overlook certain irregularities, and including both ensures comprehensive coverage.

Upon identifying the key points that provide critical details to the specified category, we proceeded to process the data. Our dataset consists of 400 training samples, each containing 20 key points. To enhance accuracy in detecting deformities, we calculated the mean and standard deviation for angles 1, 2, and 3 across all datasets for each triangle. Additionally, each triangle has associated slopes, for which we computed the mean and standard deviation of slopes 1, 2, and 3. This approach aids in identifying any potential deformities in these entities. Furthermore, we evaluated the mean and standard deviation for ratios 1 and 2 across all datasets. Initially, considering the computation of the mean for each triangle, we acknowledged that this could lead to challenges due to increased variability, which in turn could increase the standard deviation. Such an increase in variability and standard deviation could reduce sensitivity in deformity detection. Therefore, we opted for the previous method to ensure more accurate and sensitive deformity detection.

Case 1: Size-Related Irregularities

In this section, we will discuss the approach used in order to reveal abnormalities that are present in the fish structure, particularly size-related deformities. When the fish size is unusually thin or overweight, it can be considered a size-related deformity.

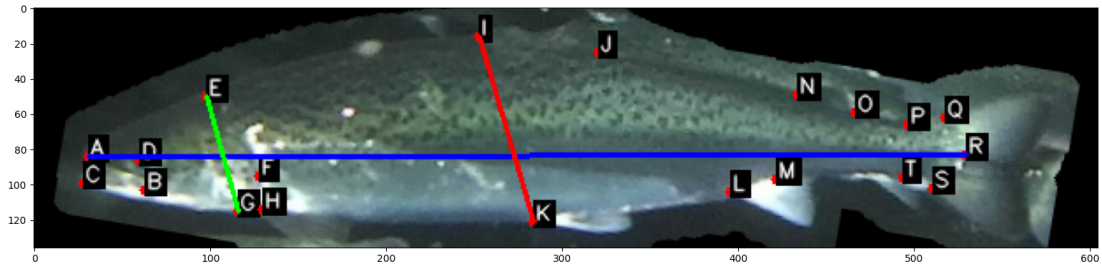


Figure 4.11: Size-Relate annotation

This irregularity can be found in different parts of the fish body, such as the head region or mid-region. To capture any size-related abnormalities that are present in the fish structure, it is important to segment the fish body into two regions: head region, mid-region, and investigate each region separately. The first region is the head region, which is the distance between the opposite key points (E to G), which is denoted green color as illustrated in this Figure 4.11. We computed the distance between E and G by using the Euclidean distance formula. The second region is mid-region which is denoted by red color as shown in Figure 4.11. It is the distance between the key points (J to K). It used Euclidean distance in order to find the distance between these key points as well. As we went further, we computed the distance between the key points (A to R), denoted in blue color as demonstrated in Figure 4.11 to create full-length to find the ratio regarding the other parameters such as head-region and mid-region. These steps are shown in Table 4.1 and also illustrated step by step in this 4.1 formula. Furthermore, we calculated the mean and standard deviation for each set of distance pairs across the entire dataset of 400 samples. For example, we determined the mean and standard deviation for the distance pairs (E-G), (J-K), and (A-R) .These values were then stored in a file in order to be used as a reference

.

Table 4.1: Baseline methodology table for thin and thickness

Distance pairs	**Explanation**	**Ratios**
A-R	Full length	N/A
E-G	Head region	$\frac{\text{full-length (A-R)}}{\text{head-region (E-G)}}$
I-K	Mid-body region	$\frac{\text{full-length}(A-R)}{\text{mid-region}(I-K)}$

$$
\begin{aligned}
\text{full-length}(A-R) &= \sqrt{(x_2-x_1)^2+(y_2-y_1)^2} \\
\text{head-region}(E-G) &= \sqrt{(x_2-x_1)^2+(y_2-y_1)^2} \\
\text{mid-region}(I-K) &= \sqrt{(x_2-x_1)^2+(y_2-y_1)^2} \\
\text{ratio 1} &= \frac{\text{full-length}(A-R)}{\text{head-region}(E-G)} \\
\text{ratio 2} &= \frac{\text{full-length}(A-R)}{\text{mid-region}(I-K)}
\end{aligned}
\tag{4.1}
$$

Case 2: Jaw Deformation

This section outlines the methodology for identifying jaw deformation, a common anomaly found in the head area of fish. We used a combination of geometric measurements, including triangles, ratios and slopes and distance pairs. The main purpose of using the **slope** 4.4 was to use it as a reference for a triangle. This approach is effective because the internal angles of the triangles might still remain constant even when it is rotated.

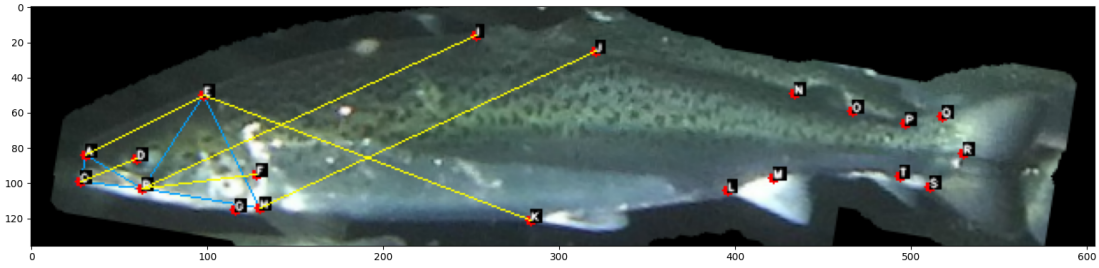


Figure 4.12: Jaw annotation with 2 triangles and their slopes

Calculating the slope between selected key points located outside the triangles and those forming the vertices of the triangles, could easily help to detect any change orientation or position. Since the slope is sensitive, if a minor positional shift in key points occurs, it can easily notify us. Then, we established a baseline that represented normal fish measurements, including angles of triangles, ratios, and slope. The baseline was extracted from a comprehensive dataset of standard fish measurements, which can be used as reference points for identifying anomalies afterward. This process is illustrated in this table 4.2. We formed two triangles from key points of (B, E, G) as the first triangle and (A, B, C) as the second triangle, and each key points that form the triangle has a connection with other key points outside of the triangle. For instance, Triangle 1 is associated with vertices (J, K, I). After identifying the connection between each vertex that formed triangles and other vertices outside, we found the slope between them as shown in Table below 4.2 and visualized in **Figure** 4.12. In order to make a precision analysis of the triangle's properties, we need to follow the following steps. With regard to this category, we have decided to use two triangles. The first step was to identify the vertices that formed the triangles, which are crucial in order to determine the shape and the size of the triangle. After identifying, we need to calculate the side length by utilizing the Euclidean Distance formula 4.3. Furthermore, we computed the angles using the Law of Cosines 4.2 and the side length that we obtained in the previous step. After these two steps, we then calculated the ratio while using the side length as demonstrated in this table 4.2. For each triangle, slopes are calculated by using

the predefined key points as illustrated in this table 4.2

$$a^2 = b^2 + c^2 - 2bc\cos A \tag{4.2}$$

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{4.3}$$

$$m = \frac{y2 - y1}{x2 - x1} \tag{4.4}$$

Table 4.2: Baseline table for Jaw using triangle's angles, slop and ratios

Triangles	**Slope**	**Ratios**
(B, E, G)	(B,J),(E,K),(G,I)	$\frac{\text{length (B-E)}}{\text{length (E-G)}}, \frac{\text{length (E-G)}}{\text{length (G-B)}}, \frac{\text{length (G-B)}}{\text{length (B-E)}}$
(A, B, C)	(A,D),(B,D),(C,D)	$\frac{\text{length (A-B)}}{\text{length (B-C)}}, \frac{\text{length (B-C)}}{\text{length (C-A)}}, \frac{\text{length (C-A)}}{\text{length (A-B)}}$

However, the Triangle's angles, slope, and ratios might not capture each deformity in the jaw. Therefore, it is essential to apply distance pairs to increase the method's robustness and capture the overlooked deformity.

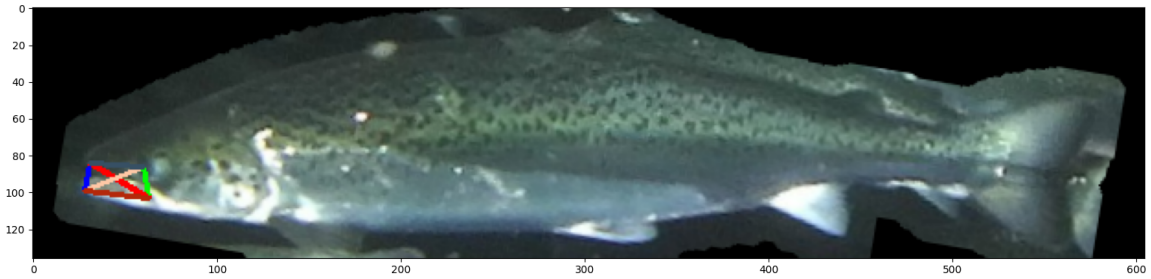


Figure 4.13: Jaw distance annotation on the fish jaw

The idea of having the distance pairs is that when vertices that make the pair of key points make any shift orientation, it will affect all other vertices that have a connection as we illustrated in **Figures 4.14** and this Table 4.13. For example, if 'A' shifts, it will affect all

the edges connected to it, and similarly, if the other key points move, they all affect each other.

Distance pairs
A-C
B-D
A-D
C-B
A-B
C-D

Table 4.3: Distances Pairs and their connection each other

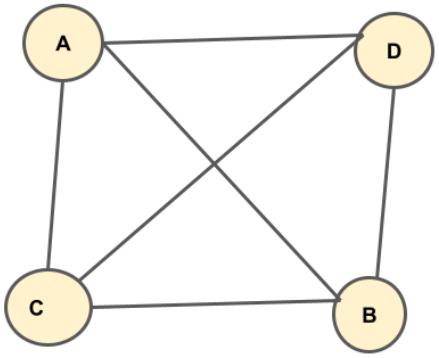


Figure 4.14: Distances Pairs and their connection each other

Case 3: Spine Deformation

The goal of a systematic approach was to identify and analyze spinal deformities in Salmon fish. Our research established a set of baseline metrics using the steps we explained in this section 4.4.2 while using these unique triangles, slopes, and ratios as outlined in Table 4.4. This table plays a crucial part in our methodology and represents the most effective pairs that can capture the presence of spine deformity. The selection of triangles and slopes was driven by their effectiveness in representing the structural aspect of spine deformity.

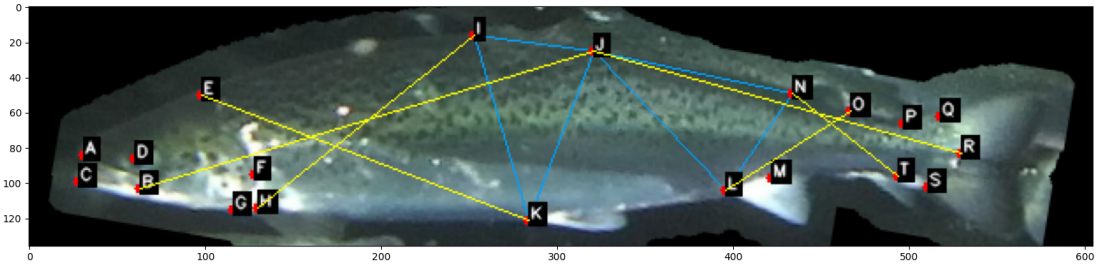


Figure 4.15: Spine annotation

Geometric pairs were meticulously chosen after experimentation and pinpointing their ability to capture wide details about the spinal abnormalities. First, we identified the vertices that could form the triangles. The second step is calculating distances between the vertices that form the triangles and then computing angles using the obtained side length while using the Law of Cosine 4.2. Furthermore, we calculate the ratio between the side lengths of the triangles. The change that occurs to the side length could indicate deformations or distortions. Lastly, we calculate the slope of each triangle in order to keep track of the triangle's orientation. The internal angles of the triangles might still remain constant even if rotated, so it was useful to establish a connection with other key points located outside of the triangle.

Table 4.4: Baseline for Spine using triangle's angles, slope and ratios

Triangles	**Slope**	**Ratios**
(I, J, K)	(I,H),(J,B),(K,E)	$\frac{\text{length (I-J)}}{\text{length (J-K)}}$, $\frac{\text{length (J-K)}}{\text{length (K-I)}}$, $\frac{\text{length (K-I)}}{\text{length (I-J)}}$
(J, L, N)	(J,R),(L,O),(N,T)	$\frac{\text{length (J-L)}}{\text{length (L-N)}}$, $\frac{\text{length (L-N)}}{\text{length (N-J)}}$, $\frac{\text{length (N-J)}}{\text{length (J-L)}}$

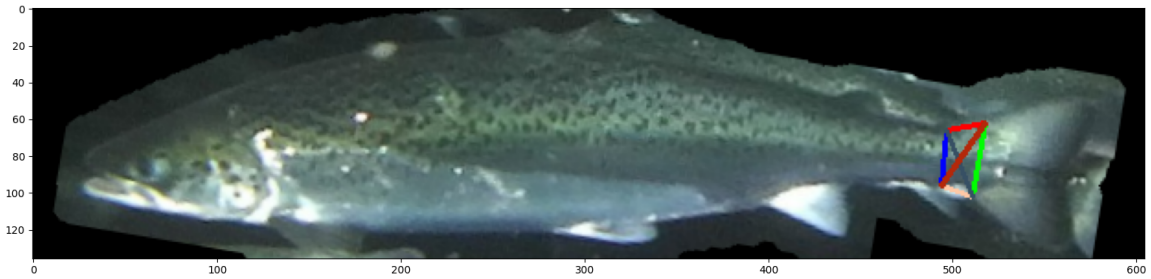


Figure 4.16: Spine distance annotation on the fish spine

Distance pairs
P-T
Q-S
P-Q
T-S
P-S
T-Q

Table 4.5: Distances Pairs and their connection each other

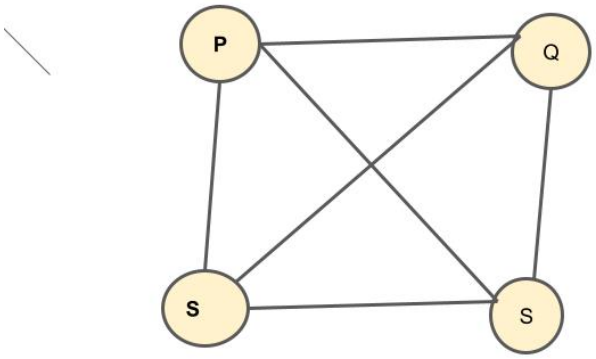
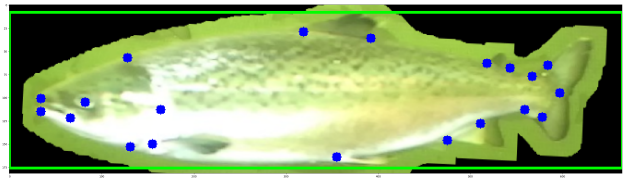


Figure 4.17: Distances Pairs for spine and their connection each other

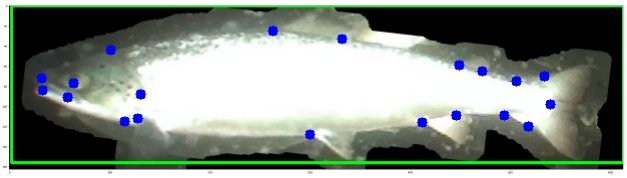
Chapter 5

Result and Discussion

In this section, we will discuss the results we have achieved by using the two models, the R-CNN keypoint detection and the YOLO model. Both models were trained over 100 epochs using 400 annotated images for training and 69 for testing. For R-CNN model training, the evaluation metrics demonstrated notable overlap with the ground truth, yielding an average precision of 1 for bounding boxes and 0.976 for key points regarding IoU threshold of 0.50. While using IoU threshold of 0.5:0.95, for bounding box the mean average precision is 0.934 and mean average precision for key point is 0.960 . Average recall also stood out, reaching up to mAR of 0.948 and 0.976 for bounding boxes and key points, respectively. The evaluation metrics were determined by considering varying IoU thresholds from 0.50 to 0.95, as illustrated in this Table 5.1.



(a) Detected key points using RCNN key point detection



(b) Detected key points using RCNN key point detection

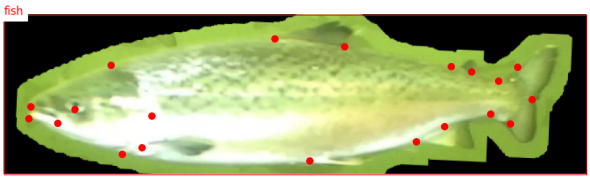
	Bounding Box (bbox)		**key points**	
Metric Description	**IoU: 0.50**	**IoU: 0.50:0.95**	**IoU: 0.50**	**IoU: 0.50:0.95**
mAP	1.000	0.934	0.976	0.960
mAR	0.960	0.948	0.985	0.976

Table 5.1: R-CNN key point detection: Performance Metrics for Bounding Box (bbox) and key points

Similarly to our approach with the R-CNN key point model, we trained the YOLO model over 400 epochs and showed impressive results. The decreasing loss metrics suggest that

the model learning process is quite efficient, as shown in Figure 5.2 below. The precision metrics for bounding boxes and key points seem close to 1, indicating a high proportion of accurate detection. Similarly, recall metrics have a value of approximately 1, indicating the model's efficacy in identifying mostly actual objects. The model achieved high mean average precision (mAP) scores, particularly with mAP50 and mAP50-95 metrics for both the boundary box and key points using different thresholds of intersection over Union (IoU).

(a) Detected key points using YOLO model



(b) Detected key points using YOLO model

Metric Description	Bounding Box (B)	key points (P)
Precision	0.999	0.999
Recall	1.000	1.000
mAP50	0.995	0.995
mAP50-95	0.995	0.983

Table 5.2: YOLO: Performance Metrics for B (Bounding Box) and P (key points)

Our tests with R-CNN and YOLO models led us to some interesting findings. YOLO scored higher in average mean precision (mAP50-95). We tested both on external images, looking specifically at their performance in key point detection and alignment. While R-CNN stood out for accurate key point alignment, as shown in the **Figures 5.17a** and **5.17b**. YOLO excelled in detecting key points but trailed behind R-CNN in alignment accuracy as shown in **Figures 5.2a** and **5.2b**. This might be because R-CNN finds the regions of interest (ROIs), classifies them and predicts their boundaries, thereby enhancing the precision of localization. On the other hand, YOLO offers fast and efficient operations. However, it compromises slightly on precision, especially when a more accurate location is needed, such as key point detection [32] [28].

.

5.1 Literature Review: Comparison of results with Other researches

The main goal of this study [23] was to evaluate the fish length through specified key points. In the study, it was used 1576 annotated images, and each fish has nine key points. They have also used the Faster R-CNN model for this purpose. The key point detection result was various and depended on using different Intersection over Union (IoU) thresholds. Using

0.5, they achieved a high accuracy of 91.7%, but as they increased the threshold of 0.95, the accuracy dropped to 85.1%. In our study, different IoU thresholds were also employed. Setting the threshold at 0.5 yielded an average precision (Ap) of 97.6%. However, incrementally increasing the threshold from 0.5 to 0.95 led to a minor decrease in accuracy, which became 96%. This suggests that our approach improves key point detection accuracy.

5.2 Geometrical Analyses

In this section, we will explain the results that we achieved using both the first approach and the second approach. Approach 1, which was used in the project's initial phase used method that incorporates multiple parameters and goes through 15 triangles and 42 distances in order to detect any deformity across the entire fish body. This is illustrated in Figure 5.3, which highlights red lines indicating where deformity occurs, while Figure 5.4 shows no deformity detected on the fish and highlights blue lines instead of red.

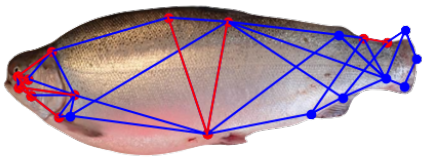


Figure 5.3: Using Approach 1 Detected Deformity

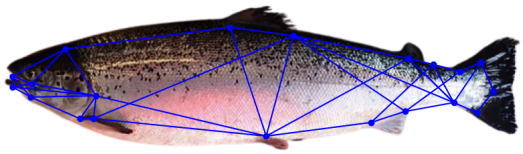


Figure 5.4: Using Approach 1 Detected No Deformity

This approach could reveal if the fish had any deformation. However, it has its own challenges as it has too many parameters, which can easily affect the result. This means the more parameters there are, the more doubt it creates. For example, a distance pair can claim a deformity in a certain place, while the triangle at that place might not indicate or detect any deformity and vice versa. Hence, it is hard to determine which method is reliable or if there is any deformity in that body part of the fish. Hence, we mainly used approach 2, which gave us more solid results and could pinpoint exactly where the deformity occurs. It also has fewer parameters, which helps reduce complexity.

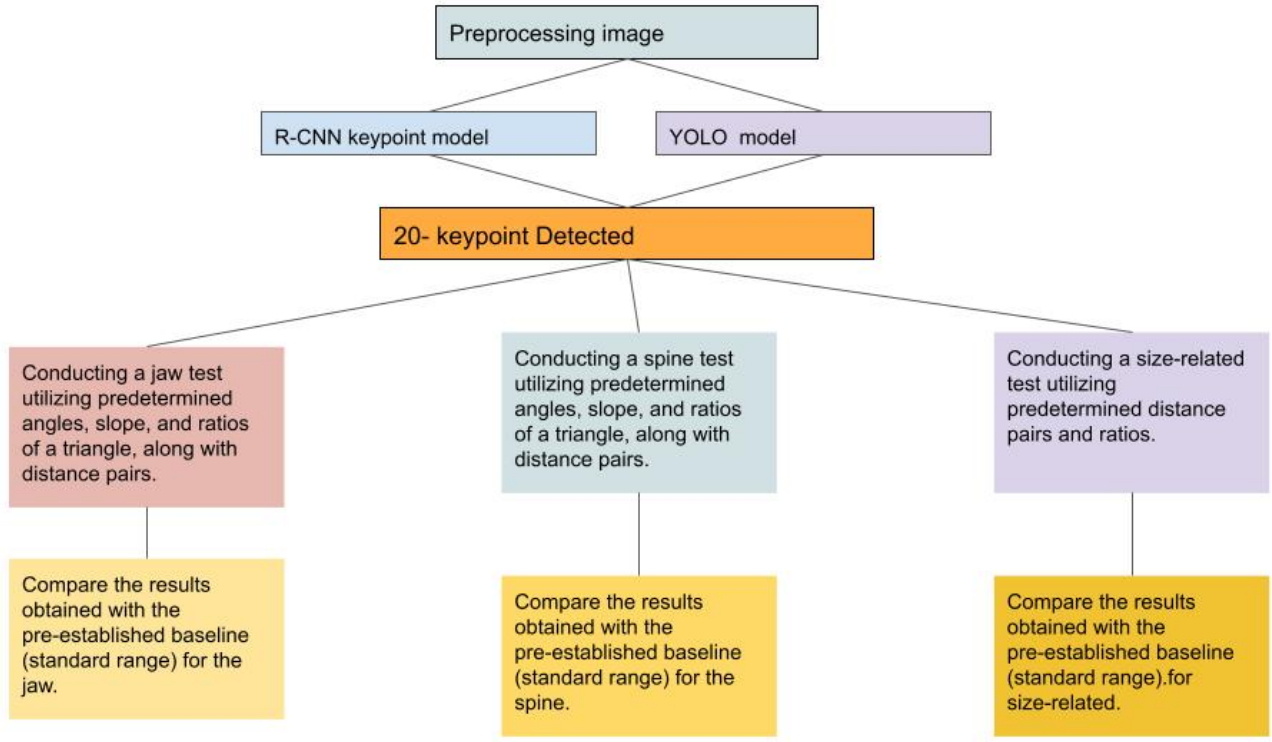


Figure 5.5: workflow

As shown in the figure above 5.5, before performing any deformity test, the image needs to undergo preprocessing to remove background noise, enhancing the accuracy of keypoint detection. Subsequently, the processed image is fed into the models for keypoint detection. The 20 keypoints obtained are then used to compute pre-defined angles, slopes, ratios, and distance pairs. These computed measurements are compared to the standard range (baseline), and any deviation from it indicates irregularity.

5.3 Case 1: Size-Related Irregularities Results

We used the method that we established in this **section 4.4.2** in order to detect the size-related irregularities that could be found in different parts of the fish body. When the full length of the fish is small, but the width is large, for example, this fish 5.6 can be easily detected due to a significantly reduced length-to-width ratio. Applying the procedure mentioned in **Table** 4.1, we calculated and set the lower and upper boundaries using **"mean ± 2 * standard deviation"** and stored them in **Table** 5.3 to compare the new keypooint measurements. We tested following fish Figure 5.6., which went through a preprocessing step in order to remove background noise and achieved this result 5.4.

Table 5.3: Baseline Size-related for Normal Fish: Mean Values with ± 2 Standard Deviations

	Lower Limit	Upper Limit
Ratio 1	5.76	8.38
Ratio 2	3.68	4.89

According to the baseline 5.3, Ratio 1 is a range of 5.756 as the lower limit and 8.38 as the upper limit. The newly detected key points were found to have a notable deviation from the baseline because the Ratio 1 value is 2.0234, laid out of the baseline Ratio 1 range. Furthermore, Ratio 2 of the baseline has a range of 3.681 as the lower limit and 4.89 as the upper limit, whereas the newly detected key points of Ratio 2 have a value of 1.015, lying outside of the baseline range. These discrepancies in both ratios strongly indicate that the new fish have size deformations, corroborated by comparing the result with baseline data.

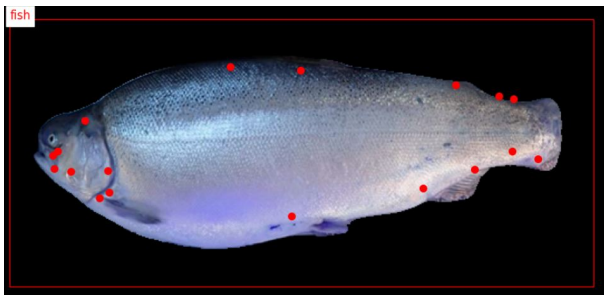


Figure 5.6: Abnormal Thickness and Size Deformities

Table 5.4: Detailed Results of Deformity Analysis in Fish 5.6

	Deformity in Ratio 1	Deformity in Ratio 2
	2.02	1.02

Furthermore, as our collection of fish images, particularly those displaying size-related anomalies, expanded, we embarked on an empirical investigation to evaluate the significance of our observations. For instance, the fish shown in 5.7 displays distinct size-related deformities, exemplifying unusual thinness compared to normal fish. The table 5.3

that we established outlines the lower and upper limit for the normal fish, compared to this finding 5.5 of newly detected key points that belong to this fish **Figure** 5.7. The comparison shows that the fish have a deformity. Again, this shows how reliable this method is.

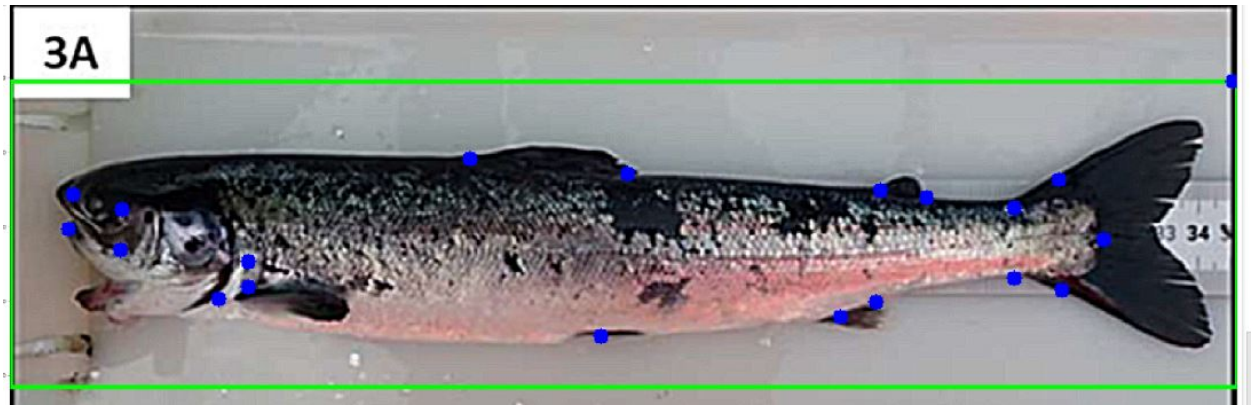


Figure 5.7: This fish has a type of thin deformity.

Table 5.5: Deformity Data for Thin and Thickness

	Deformity in Ratio 1	Deformity in Ratio 2
	4.75	5.828

Moreover, tests conducted on the fish shown in 5.8 revealed significant deformations. Specifically, in the case of Ratio 2, the established standard range lies between 3.68 and 4.89. However, for this particular fish, Ratio 2 was recorded at 5.604, thereby confirming the existence of a size-related irregularity. Additionally , the results can precisely point the location of deformities. In this particular case, the deformity is confined to the mid-region of the salmon's body. As explained in Table 4.1, the segmentation of the salmon body is divided into two regions. The head region, defined by the distance pair E-G, corresponds to ratio 1, while the mid-region, determined by the distance pair I-K, corresponds to ratio 2. With regard to the standard range 5.3, ratio 1 falls within the standard range, whereas ratio 2 is beyond the normal range.

Overall, this methodology demonstrates its reliability in the identification of size-related irregularities, including unusual thinness and thickness.

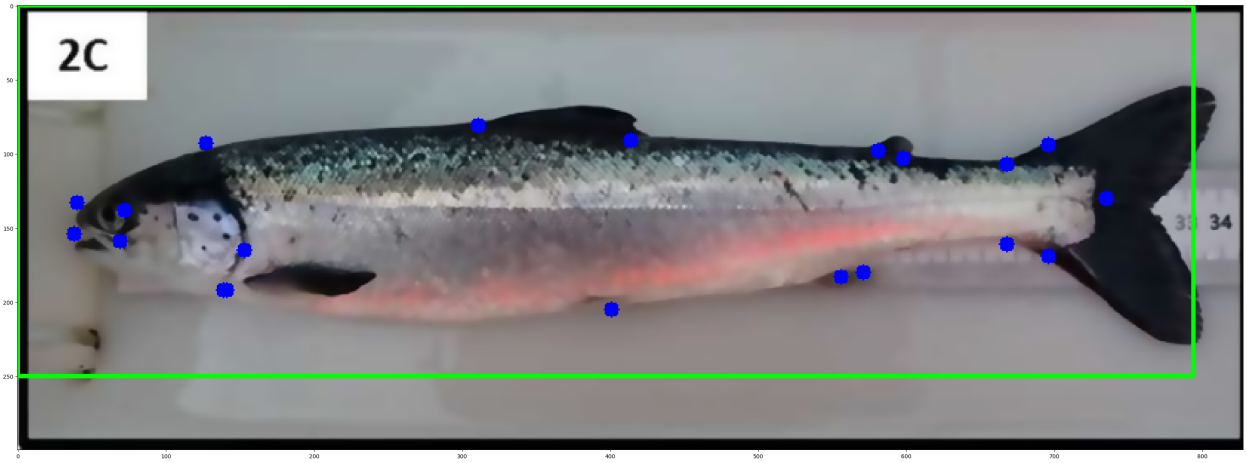


Figure 5.8: Thin deformity

Table 5.6: Deformity Data for Thin and Thickness

	No Deformity in Ratio 1	Deformity in Ratio 2
	7.020	5.604

5.3.1 Case 2: Jaw Deformity Result

In our study, we selected significant pairs of triangles, slopes and distances pairs that could provide critical insight into the fish's jaw structure, which were vital for identifying jaw deformation. The features were extracted using the pairs, and then the mean and standard deviations were computed. The most important step was determining a threshold; it required experimentation, and we used a **"mean ± 2 * standard deviation"** for the lower and upper limits. The findings are presented in Table 5.7 and Table 5.8.

*Table 5.7: Baseline for Jaw using triangle, ratio and slope using mean ± 2 * standard deviation*

	Lower Limit (T1)	Upper Limit (T1)	Lower Limit (T2)	Upper Limit (T2)
Angle 1	33.74	59.14	8.42	30.54
Angle 2	42.92	65.16	76.35	134.11
Angle 3	61.19	97.83	29.41	81.17
Slope 1	-0.43	-0.19	-0.22	0.46
Slope 2	0.29	0.57	Inf	Inf
Slope 3	-1.01	-0.57	-0.84	0.24
Ratio 1	0.69	0.97	1.04	1.32
Ratio 2	0.99	1.75	2.19	2.95
Ratio 3	0.70	1.10	0.15	0.55

*Table 5.8: Baseline for Jaw using distances pairs using mean ± 2 * standard deviation*

Distance Pairs	Mean	Std	Lower Limit	Upper Limit
A-C	11.97	3.03	5.90	18.04
B-D	16.74	4.08	8.57	24.92
A-D	34.27	6.65	20.95	47.58
C-B	33.23	11.86	9.51	56.96
A-B	32.81	11.55	9.69	55.93
C-D	29.27	6.56	16.14	42.39

When the model detects new key points, it computes the triangles, slopes, ratios, and distance pairs using established pairs as we explained in these tables Figure 4.14 and Table 4.2. These are then compared against the baseline data. Deviation from the baseline range, as outlined in our tables, will demonstrate jaw deformation. We used two different tables because triangle properties might overlook some jaw deformity such as this fish's jaw deformity Figure 5.10. Hence, distance pairs address irregularities potentially missed by the triangular properties.

Using pre-established **Tables** 5.7 ,5.8, we reviewed them to test how effectively these methods work. Initially, we feed the fish 5.9 into the model to detect the key points. Having a key point in place, we could efficiently determine triangular properties, such as finding the triangles' predetermined angles, slopes, and ratios, as shown in this table 4.2. Noteworthy is that this fish figure 5.9 has both jaw- and size-related anomalies, but we exclusively concentrate on jaw deformity. Then, We compared the finding of the newly detected key point of this fish 5.9 into the baseline Table 5.7, and we can clearly observe that the slope 1 and 2 of the baseline for triangle 1 was range set between -0.43 and -0.19, and slope 2 was between 0.29 and 0.57 while the result of the newly detected fish for slope 1 is -0.155 and slope 2 is 0.58. This deviation is a clear suggestion that this fish has jaw deformity.

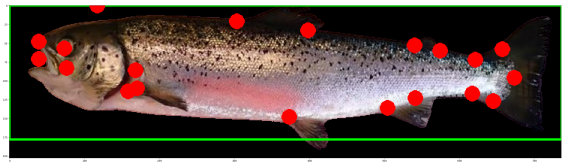


Figure 5.9: Detected Jaw Deformity

Table 5.9: Newly Detected Jaw Deformity Result

	Triangle 1	Triangle 2
Angle 1	46.074	25.44
Angle 2	49.51	107.97
Angle 3	84.42	46.59
Slope 1	-0.155	0.26
Slope 2	0.58	8.67
Slope 3	-0.64	-0.41
Ratio 1	0.76	1.31
Ratio 2	1.38	1.69
Ratio 3	0.95	0.45

In our experiment, we introduced distance pairs as an additional method to detect jaw deformities in fish, as detailed in Table 5.8. This approach was adopted after the initial method was not completely reliable due to ignoring certain abnormalities that is found in the fish. The integration of distance pairs has proven more effective, particularly in instances where the initial approach failed to identify deformities, as demonstrated in Figure 5.10. Upon comparison of the observed values in Table 5.11 with the baseline data in Table 5.8, it was found that the A-B pair is outside the established range. This deviation strongly suggests a deformity in the fish's jaw. This findings suggests that using distance pairs with triangular properties methods could help to improve the identification of the anomalies that exist in the fish. Relying only on the first method as we did in our first experimentation, could potentially ignore certain deformities such as the deformity that exists in this fish in Figure 5.10.

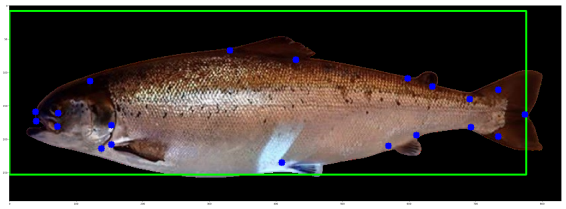


Figure 5.10: Detected Jaw Deformity

Figure 5.11: Result that was obtained using distance pairs

	A-C	B-D	A-D	C-B	A-B	C-D
	14.03566885	20	22	12	2	18

5.3.2 Case 3: Spine Deformity Result

This section will analyze the spin deformity identified as one of the categories we mentioned previously. This type of abnormality is frequently observed in a fish's mid and terminal regions. To facilitate comparative analysis, we established a baseline for this category. Our methodology includes using triangles, slopes, and ratios and distances pairs as illustrated in Table 5.10 and this Table 5.11.

*Table 5.10: Baseline for Spine using triangle, ratio and slope using mean ± 2 * standard deviation*

	Upper Limit (T1)	Lower Limit (T1)	Upper Limit (T2)	Lower Limit (T2)
Angle 1	94.48	67.64	86.41	60.01
Angle 2	44.42	23.02	83.07	59.15
Angle 3	74.84	55.6	52.12	27.46
Slope 1	-0.602	-1.118	1.396	0.664
Slope 2	1.396	0.644	-0.13	-0.57
Slope 3	0.574	0.286	1.158	0.542
Ratio 1	0.796	0.428	2.007	1.263
Ratio 2	1.107	0.829	0.764	0.464
Ratio 3	2.414	1.218	1.147	0.879

*Table 5.11: Baseline for spine using distances pairs using mean ± 2 * standard deviation*

Distance Pairs	Mean	Std	Lower Limit	Upper Limit
P-T	37.578533	6.221585	25.135364	50.021702
Q-S	57.585282	8.303778	40.977725	74.192838
P-Q	28.569675	5.253109	18.063456	39.075894
T-S	28.238751	5.905517	16.427717	40.049785
P-S	52.206498	7.716079	36.774340	67.638657
T-Q	56.632649	9.028666	38.575317	74.689982

We computed the lower bound and upper bound measurements for these metrics. This baseline was used as a comparative analysis to identify spine deformity. If the newly detected key points value falls outside these ranges, this could be identified as spine deformity. We feed the model the image of the fish below after it went through preporcessing such as removing background noise, and then detecting key points as illustrated in Figure 5.14. Initially, we examined the key points using triangle properties such as angles, ratio, and slope using detected key points. The result is shown in this Table 5.13.

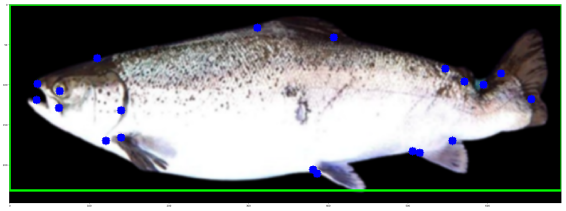


Figure 5.12: Detected Spine Deformity

Figure 5.13: Result that obtained from detected key points with spine deformity

	Triangle 1	Triangle 2
Angle 1	88.17	53.819
Angle 2	30.53	88.617
Angle 3	61.29	37.563
Slope 1	-0.8011	0.31048
Slope 2	-0.2550	-1.589
Slope 3	0.5129	-2.512
Ratio 1	0.57	1.64
Ratio 2	0.87	0.76
Ratio 3	1.97	0.807

When we compared the result to baseline data, it's evident that for triangle 2, angles 1, 2, and 3 with values of 53.82, 88.62, and 37.56 fall outside the normal ranges. The baseline for triangle 2, angle 1 is (60.01,84.41), angle 2 (59.15,83.07) and angle 3 (27.46.46,52.12). This comparison clearly shows that the fish have spine deformity.

To prove this, we have also used distance pairs. It also indicated the presence of anomalies in this fish. As illustrated in this **Table** 5.11 compared to **Table** 5.13.

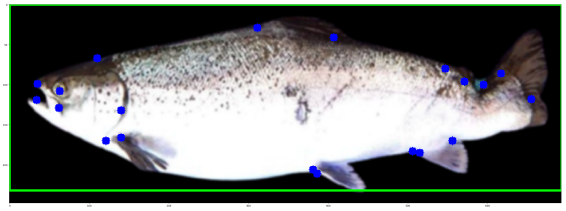


Figure 5.14: Detected Spine Deformity

Table 5.12: 2 standard deviation

	P-T	Q-S	P-Q	T-S	P-S	T-Q
	121.69	262.65	26.07	123.22	236.647	147.41

Table 5.13: Result that was obtained using distance pairs for spine

This result was obtained using distance pairs for spine, shown in Table 5.13, shows a more considerable deviation from the standard range. This could be attributed to the fish displaying more significant spinal damage, especially towards the end of the spine, causing the key points to shift from their original positions, and it can also lead to increased tightness. This could cause some key points to relocate and potentially hinder accurate detection.

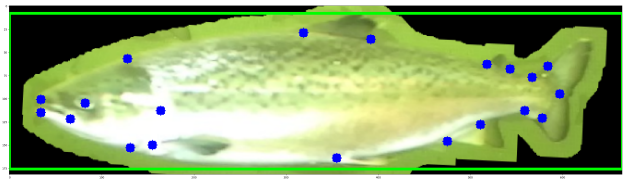
Figure 5.15: Detected Spine Deformity

Figure 5.16: Result obtained from detected key points with spine deformity

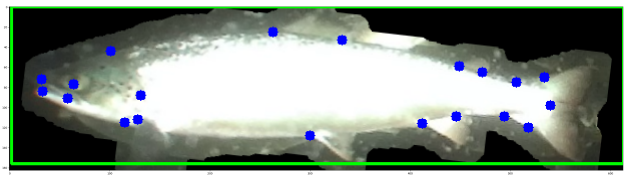
	Triangle 1	Triangle 2
Angle 1	87.05	60.25
Angle 2	30.91	42.04
Angle 3	62.03	77.69
Slope 1	-0.9716	0.3277
Slope 2	-0.3403	-0.6528
Slope 3	0.4408	4.3
Ratio 1	0.5816	0.6854
Ratio 2	0.8844	1.1253
Ratio 3	1.9440	1.2964

The presented Table 5.16 demonstrates the irregularities present in the fish illustrated in Figure 5.15. Upon comparison with established standard ranges, it becomes evident that the ratios of Triangle 2 exceed the standard range. Furthermore, angles 2 and 3 of Triangle 2 also fall out of the standard range. Notably, the associated slope 2 of Triangle 2 deviates from the expected range.

Finally, we tested a normal fish with no deformity taken from the test dataset to verify that our established methodology works across all fish. This test dataset was not used as part of establishing our baseline. As shown in Figure 5.17a 5.17b. This shows how well this geometric analysis with key point detection identifies and reveals deformities in the Salmon fish structure. As we showed in the previous **section** 5.3.2 5.3.2, 5.3, we have established a baseline for each category, which helped us analyze each category more precisely and with high accuracy.



(a) Normal fish



(b) Normal fish

5.4 Challenges

In the course of this project, we faced various challenges. The main challenge was due to the limited size of the datasets. With only 469 samples available, the dataset was insufficient for the complexity of the task. This limitation leads to the lack of adequate variation between the fish sizes. Because the datasets do not present the full spectrum of fish size variation. Consequently, setting up an appropriate threshold for analysis became a difficult

task. Furthermore, the quality of the test images imposed additional challenges. Many real world images contained extra background and noises, which complicates the ability of the model to detect key points accurately. To tackle this challenge, we implemented a system that could automatically remove background and noise, but it was not enough, so it required manual removal; therefore, we used online software. However, when removing background from an image, it is carried out with the risk of removing partial information that is important from the image and could have an undesirable outcome.

Chapter 6

Conclusions

6.1 Conclusion

This thesis explores the relationship between the cutting-edge technologies such as artificial intelligence and its major impact in aquaculture. The main focus of this project is to delve into parts of AI such as keypoint detection with a combination of geometrical analysis to present a solution for detecting irregularities in salmon fish in aquaculture. We have employed a dataset that contains 469 annotated images, where each image contains 20 key points. In this study, R-CNN keypoint detection and YOLO models were utilized. Both models were trained over a good amount of epochs and have shown impressive results. The R-CNN model excelled in keypoint detection accuracy and alignment, while the YOLO model achieved high average precision (AP) accuracy. On the other hand, two different approaches were used in order to identify deformity in aquaculture salmon fish. The initial phase of this project was to form 15 unique triangles and 42 distance pairs to conduct a geometrical analysis to gain details on the fish structure and if any deformities were present in the fish. However, we did not process this approach due to the high number of parameters, which led to uncertainties. Hence, we came up with a second approach with fewer parameters and a tidier way of classifying deformity into three different cases. The cases include jaw deformity, size-related (thin or thick fish), and spinal deformity. This approach uses geometric analysis such as triangles, ratios, slopes, and distances. Then, the mean and standard deviation of each of these were computed. This helps to establish a baseline, which we used as a reference in order to compare to the newly detected key point. Any deviation from the baseline could be an indication of irregularity . This approach has been shown to be much more accurate than the first approach and detects irregularities more precisely.

In contrast to the findings of another study, our research in keypoint detection reveals notable differences. The results we mentioned in section 5.1 have acheived 85.1% accuracy while our

result was 96% accurarcy of mean average precision (AP).
Overall, this thesis illustrates a significant step forward in addressing the complex challenges in aquaculture fish particularly in Salmon fish. Our findings have demonstrated a noteworthy improvement in the accuracy and precision of fish deformity detection and have established a new practice in the field by bridging the gap between cutting-edge technology and traditional aquaculture practices.

6.2 Future Work

Even after achieving the goal set for this thesis, there is still much more to do in order to create a system that is able to identify correctly irregularities in the fish. Things that need to be considered are datasets that contain fish of different sizes. The current datasets contain small samples, which cannot accurately represent the overall structure of the fish. Increasing datasets and having high variance could help to set a threshold, for instance, using the current dataset, was a complex task to find a good threshold because the dataset was insufficient and required a lot of adjustment to reach a good threshold.In order to increase the variability of the image and datasets, the use of an Adversarial Generative Network might be a good candidate. It enables the generation of a new Salmon image from the existing one. Furthermore, another thing that needs to be worked on is preprocessing datasets. In real-world scenarios, the images have background and noises, which can downgrade the performance of the model. Thus, to achieve a desirable result, it could be useful to have a good mechanism for preprocessing images that tackle all sorts of image backgrounds and noises. Using a simple program for preprocessing can well work for the short term but might not be suitable for the long term. Because the image can be captured under various condition and dealing with these conditions require much more time with care and consideration.
Finding triangles and slopes that could give us more details about the structure of the fish was based on experiment, it could be useful to utilize an algorithm that can find the pairs that capture the vital information about the shape of the fish. We might suggest such a Genetic algorithm that can ensure to finding of the pairs that work best with less amount of time.

Bibliography

[1] *(PDF) A Survey on Performance Metrics for Object-Detection Algorithms*. https://www.researchgate.net/publication/343194514_A_Survey_on_Performance_Metrics_for_Object-Detection_Algorithms. (Accessed on 12/17/2023).

[2] *[1703.06870] Mask R-CNN*. https://arxiv.org/abs/1703.06870. (Accessed on 12/27/2023).

[3] *1512.03385.pdf*. https://arxiv.org/pdf/1512.03385.pdf. (Accessed on 10/10/2023).

[4] *1804.06208.pdf*. https://arxiv.org/pdf/1804.06208.pdf. (Accessed on 10/10/2023).

[5] *A Complete Guide to Data Augmentation | DataCamp*. https://www.datacamp.com/tutorial/complete-guide-data-augmentation. (Accessed on 10/18/2023).

[6] *A Gentle Introduction to Object Recognition With Deep Learning - MachineLearningMastery.com*. https://machinelearningmastery.com/object-recognition-with-deep-learning/. (Accessed on 12/30/2023).

[7] *A Gentle Introduction to the Rectified Linear Unit (ReLU) - MachineLearningMastery.com*. https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/. (Accessed on 12/30/2023).

[8] *AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the difference?* https://www.ibm.com/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks/. (Accessed on 10/18/2023).

[9] *An introduction to object detection with deep learning - TechTalks*. https://bdtechtalks.com/2021/06/21/object-detection-deep-learning/. (Accessed on 09/06/2023).

[10] *Analyzing 6 Types of Neural Networks in Deep Learning*. https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/. (Accessed on 10/18/2023).

[11] *Artificial Intelligence in Aquaculture Farming | NeuroSYS*. https://neurosys.com/services/ai-in-aquaculture-farming. (Accessed on 10/18/2023).

[12] *Deep Learning for Object Detection: A Comprehensive Review | by Joyce Xu | Towards Data Science*. https://towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9. (Accessed on 09/06/2023).

[13] *Deep Learning Techniques—R-CNN to Mask R-CNN: A Survey | SpringerLink*. https://link.springer.com/chapter/10.1007/978-981-13-9042-5_56. (Accessed on 09/05/2023).

[14] Junyu Dong et al. "A detection-regression based framework for fish keypoints detection." In: *Intelligent Marine Technology and Systems* 1.1 (2023), p. 9.

[15] *Fish Keypoints Detection for Ecology Monitoring Based on Underwater Visual Intelligence | IEEE Conference Publication | IEEE Xplore*. https://ieeexplore.ieee.org/document/9305424. (Accessed on 12/17/2023).

[16] *Fishes | Free Full-Text | Accurate Wound and Lice Detection in Atlantic Salmon Fish Using a Convolutional Neural Network*. https://www.mdpi.com/2410-3888/7/6/345. (Accessed on 10/16/2023).

[17] *GitHub - rafaelpadilla/Object-Detection-Metrics: Most popular metrics used to evaluate object detection algorithms*. https://github.com/rafaelpadilla/Object-Detection-Metrics. (Accessed on 10/14/2023).

[18] *GitHub - ultralytics/ultralytics: NEW - YOLOv8 in PyTorch > ONNX > OpenVINO > CoreML > TFLite*. https://github.com/ultralytics/ultralytics. (Accessed on 12/17/2023).

[19] *Home - Ultralytics YOLOv8 Docs*. https://docs.ultralytics.com/#yolo-a-brief-history. (Accessed on 12/17/2023).

[20] *Human Pose Estimation using Keypoint RCNN in PyTorch*. https://learnopencv.com/human-pose-estimation-using-keypoint-rcnn-in-pytorch/. (Accessed on 10/10/2023).

[21] *Intersection Over Union IoU in Object Detection Segmentation*. https://learnopencv.com/intersection-over-union-iou-in-object-detection-and-segmentation/. (Accessed on 10/13/2023).

[22] *Introduction to Object Detection Algorithms*. https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/. (Accessed on 09/06/2023).

[23] *Key point detection method for fish size measurement based on deep learning - Yu - 2023 - IET Image Processing - Wiley Online Library*. https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/ipr2.12924. (Accessed on 12/28/2023).

[24] *mAP : Evaluation metric for object detection models | by David Cochard | axinc-ai | Medium*. https://medium.com/axinc-ai/map-evaluation-metric-of-object-detection-model-dd20e2dc2472. (Accessed on 10/13/2023).

[25] *Mask RCNN with RESNET50 for Dental Filling Detection*. https://thesai.org/Downloads/Volume12No10/Paper_79-Mask_RCNN_with_RESNET50_for_Dental_Filing_Detection.pdf. (Accessed on 12/17/2023).

[26] Umar Farouk Mustapha et al. "Sustainable aquaculture development: a review on the roles of cloud computing, internet of things and artificial intelligence (CIA)." In: *Reviews in Aquaculture* 13.4 (2021), pp. 2076–2091.

[27] *Object Detection Explained: R-CNN | by Ching (Chingis) | Towards Data Science*. https://towardsdatascience.com/object-detection-explained-r-cnn-a6c813937a76. (Accessed on 12/27/2023).

[28] *Object Detection Explained: R-CNN | by Ching (Chingis) | Towards Data Science*. https://towardsdatascience.com/object-detection-explained-r-cnn-a6c813937a76. (Accessed on 12/27/2023).

[29] *paper_survey_on_performance_metrics_for_object_detection_algorithms.pdf*. file:///C:/Users/Bruker/Downloads/paper_survey_on_performance_metrics_for_object_detection_algorithms.pdf. (Accessed on 10/14/2023).

[30] *R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms | by Rohith Gandhi | Towards Data Science*. https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e. (Accessed on 09/05/2023).

[31] *R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms | by Rohith Gandhi | Towards Data Science*. https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e. (Accessed on 09/05/2023).

[32] *R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms | by Rohith Gandhi | Towards Data Science*. https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e. (Accessed on 12/26/2023).

[33] *Remote Sensing | Free Full-Text | Progress in the Application of CNN-Based Image Classification and Recognition in Whole Crop Growth Cycles*. https://www.mdpi.com/2072-4292/15/12/2988. (Accessed on 10/18/2023).

[34] *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions | Journal of Big Data | Full Text*. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00444-8. (Accessed on 12/30/2023).

[35] *Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation*. https://openaccess.thecvf.com/content_cvpr_2014/papers/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.pdf. (Accessed on 09/05/2023).

[36] *Sensors | Free Full-Text | An Efficient Human Instance-Guided Framework for Video Action Recognition*. https://www.mdpi.com/1424-8220/21/24/8309. (Accessed on 10/10/2023).

[37] *Softmax Activation Function with Python - MachineLearningMastery.com*. https://machinelearningmas com/softmax-activation-function-with-python/. (Accessed on 12/30/2023).

[38] *Softmax Function Definition | DeepAI*. https://deepai.org/machine-learning-glossary-and-terms/softmax-layer#:~:text=It%20is%20clear%20from%20this,can%20be%20interpreted%20as%20probabilities.. (Accessed on 12/30/2023).

[39] Feiyang Suo et al. "Fish Keypoints Detection for Ecology Monitoring Based on Underwater Visual Intelligence." In: *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. 2020, pp. 542–547. DOI: 10.1109/ICARCV50220.2020.9305424.

[40] *Sustainable Food Supply Chain | UBS Global*. https://www.ubs.com/global/en/assetmanagement/insights/investment-outlook/panorama/panorama-mid-year-2023/articles/sustainable-food-supply.html?&campID=SEM-SIPASSIVE2-GLOBAL-ENG-GOOGLE-Panorama-Food-sustainable_farming-PHRASE-ANY&gclid=CjOKCQjwhL6pBhDjARIsAGx8D5_GuplF-r_LSZD4ZTXL0Xxw_WjtA6lkz55NZCMaVYNVUX_ZzQuNPNEaAkAeEALw_wcB. (Accessed on 10/18/2023).

[41] *The rise of AI in aquaculture | The Fish Site*. https://thefishsite.com/articles/the-rise-of-ai-in-aquaculture-artificial-intelligence. (Accessed on 10/18/2023).

[42] *The State of World Fisheries and Aquaculture 2022*. https://www.fao.org/3/cc0461en/cc0461en.pdf. (Accessed on 10/18/2023).

[43] *Underwater abnormal classification system based on deep learning: A case study on aquaculture fish farm in Taiwan - ScienceDirect*. https://www.sciencedirect.com/science/article/abs/pii/S0144860922000668?via%3Dihub. (Accessed on 12/30/2023).

[44] *What are Convolutional Neural Networks? | Definition from TechTarget*. https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network. (Accessed on 10/18/2023).

[45] *What is Data Augmentation in a CNN? Python Examples*. https://nnart.org/what-is-data-augmentation-in-a-cnn/#Data_Augmentation_in_a_CNN. (Accessed on 09/28/2023).

[46] *YOLO — You only look once, real time object detection explained | by Manish Chablani | Towards Data Science*. https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006. (Accessed on 10/18/2023).

[47] *YOLO Algorithm for Object Detection Explained [+Examples]*. https://www.v7labs.com/blog/yolo-object-detection. (Accessed on 10/18/2023).

[48] *YOLO Detection – Accelerating Non Maximal Suppression | Quadric*. https://quadric.io/2022/01/19/yolo-detection-accelerating-non-maximal-suppression/. (Accessed on 12/30/2023).

[49] *YoloV8 Pose Estimation and Pose Keypoint Classification using Neural Net PyTorch | by Ali Mustofa | Medium*. https://alimustoofaa.medium.com/yolov8-pose-estimation-and-pose-keypoint-classification-using-neural-net-pytorch-98469b924525. (Accessed on 12/17/2023).