designs

MDPI

Article

Lobster Position Estimation Using YOLOv7 for Potential Guidance of FANUC Robotic Arm in American Lobster Processing

Nawal Chelouati *, Yassine Bouslimani and Mohsen Ghribi

Robotics, Electronics & Industry 4.0 Lab., Université de Moncton, Moncton, NB E1A 3E9, Canada
* Correspondence: enc4349@umoncton.ca

Abstract: The American lobster (*Homarus americanus*) is the most valuable seafood on Canada's Atlantic coast, generating over CAD 800 million in export revenue alone for New Brunswick. However, labor shortages plague the lobster industry, and lobsters must be processed quickly to maintain food safety and quality assurance standards. This paper proposes a lobster estimation orientation approach using a convolutional neural network model, with the aim of guiding the FANUC LR Mate 200 iD robotic arm for lobster manipulation. To validate this technique, four state-of-the-art object detection algorithms were evaluated on an American lobster images dataset: YOLOv7, YOLOv7-tiny, YOLOV4, and YOLOv3. In comparison to other versions, YOLOv7 demonstrated a superior performance with an F1-score of 95.2%, a mean average precision (mAP) of 95.3%, a recall rate of 95.1%, and 111 frames per second (fps). Object detection models were deployed on the NVIDIA Jetson Xavier NX, with YOLOv7-tiny achieving the highest fps rate of 25.6 on this platform. Due to its outstanding performance, YOLOv7 was selected for developing lobster orientation estimation. This approach has the potential to improve efficiency in lobster processing and address the challenges faced by the industry, including labor shortages and compliance with food safety and quality standards.

Keywords: American lobster; machine-vision; convolutional neural networks; object orientation; Nvidia Jetson

check for updates

Citation: Chelouati, N.; Bouslimani, Y.; Ghribi, M. Lobster Position Estimation Using YOLOv7 for Potential Guidance of FANUC Robotic Arm in American Lobster Processing. *Designs* **2023**, *7*, 70. https://doi.org/10.3390/designs7030070

Academic Editors: Roveda Loris and Oscar Reinoso Garcia

Received: 4 April 2023
Revised: 18 April 2023
Accepted: 22 May 2023
Published: 23 May 2023

1. Introduction

The American lobster (*Homarus americanus*) industry relies on various transformation processes to ensure the high quality of its products. Quality assurance is crucial in this sector, as Renaud et al. [1] reported that quality factors can fluctuate due to inconsistencies in labor practices and non-existent or unenforced procedures. Moreover, the limited American lobster fishing season imposes time constraints on processing, which are not always met, thus affecting the quality of the final product [1]. Furthermore, the Atlantic Canada Opportunities Agency highlighted a critical labor and skills shortage in Atlantic Canadian businesses, with American lobster processors experiencing production limitations as more workers leave the industry [2]. As a result, many processing plants in New Brunswick struggle to process the volume of American lobster which is caught. Addressing this issue requires expanding automation and implementing vision-guided robots in processing operations [2].

In recent years, there has been significant progress in computer-based vision and deep convolutional neural network (CNN) methods for applications in agriculture [3] and food processing [4], such as detection, recognition, and segmentation. Various studies have demonstrated the effectiveness of object detection techniques, including YOLO and Faster R-CNN, in tackling challenges across food processing and agriculture sectors. These techniques have been applied to various food items, such as carrots, ham, apples, mutton, fruits, shrimp, Nile tilapia, and Atlantic salmon [5–12].

Several other studies have applied deep learning and computer vision techniques to different targets in the food industry, reinforcing the significance of our work. In the poultry industry, Liu et al. [13] proposed a real-time object detection system using YOLOv4 to identify and track chickens in a poultry house, achieving a precision of 95.24%. A CNN-based detector was used by Zhang et al. [14] to detect and segment individual pigs in images taken from pig farms, resulting in a 94.72% detection accuracy. In the dairy industry, Salau et al. [15] utilized Mask R-CNN to detect, segment, and track individual cows with an average precision of 91% and 85% for bounding boxes and segmentation masks, respectively. In addition, Zheng et al. [16] used the YOLOv3 model to detect various types of vegetables in greenhouse environments, such as tomatoes, cucumbers, and chili peppers; over 99% accuracy was achieved using their approach.

The results of these studies demonstrate the versatility and effectiveness of deep learning and computer vision techniques across a variety of food industries. By adapting these methods for the American lobster processing industry, our research contributes to the growing body of knowledge and further validates the value of such approaches in addressing labor shortages, improving product quality, and automating tasks within the sector.

This work aims to develop an automated method for processing American lobster using a fast and accurate machine vision solution. We propose a lobster orientation estimation algorithm combined with a transfer learning approach to retrain pre-trained object detectors using an American Lobster images dataset. Models are deployed on the embedded platform Nvidia Jetson Xavier NX for real-time lobster part detection. This approach offers the industry an opportunity to utilize vision-guided machines for numerous tasks currently performed manually, including exhausting, repetitive tasks and cutting operations that pose hazards to workers. This study addresses the labor shortage faced by industries in New Brunswick and holds the potential to create new jobs, attracting younger people to the seafood processing industry where, presently, the average employee age is approximately 55 years [17].

2. Materials and Methods

2.1. Robotics-Integrated Vision Systems

In recent years, there has been a growing interest in developing advanced vision systems for robotic applications. This is particularly useful for automating complex tasks such as identifying and manipulating objects of varying shapes and sizes. In a previous study [18], we investigated two distinct vision systems for enabling the FANUC robotic arm in Figure 1 to recognize and locate lobsters. The FANUC vision-based solution, the IRVision system, was assessed using two different tools: the Curved Surface Matching (CSM) Locator and the Geometric Pattern Matching (GPM) Locator.

The GPM Locator is a computer vision technology designed to identify and locate specific geometric shapes or patterns in an image. The CSM Locator is an effective computer vision solution used to identify and locate curved surfaces on an object. These solutions are often used in manufacturing and quality control applications, where it is necessary to precisely locate and inspect components or products on a production line. However, the experiments conducted in the previous study showed that both tools exhibit limitations in the detection of effectiveness and speed. They are not the most effective solutions for more complex shapes or objects that do not have a distinct geometric pattern or feature such as lobsters. In contrast, the object detection model based on the YOLOv4 algorithm showed promising results when implemented on the NVIDIA Jetson Xavier NX.

Building upon this foundation, the present study aims to further explore and enhance the capabilities of the YOLO-based vision system for lobster detection and orientation estimation. In this work, we have combined an object orientation estimation algorithm with the YOLOv7 model to improve the detection and identification of lobster body parts and their spatial orientation, evaluated the performance of this integrated approach, and assessed its effectiveness in accurately detecting lobster body parts while maintaining real-time processing speeds.

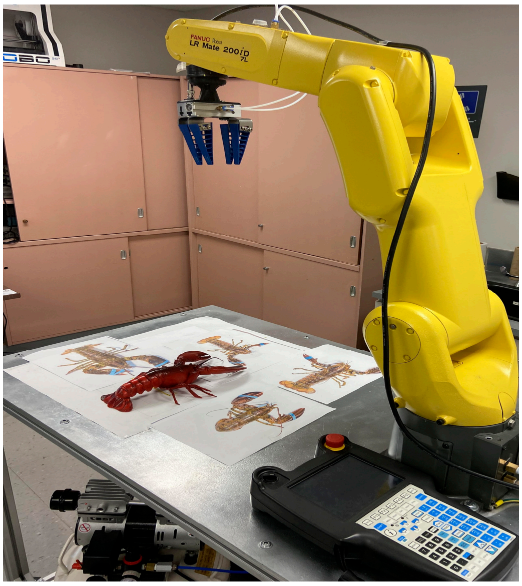


Figure 1. FANUC LR Mate 200 iD setup using iRVision.

2.2. Data Pre-Processing

As far as we are aware, there is no open-source American lobster dataset providing representative samples that can be used for developing automated object detection models for lobster position estimation. Continuing our research from [18], 1000 images of cooked lobsters of various sizes were added to the dataset. These images consider the variation in lighting conditions and lobster orientation. Furthermore, to avoid the problem of overfitting, the number of images is increased through data augmentation. This paper adopts a data augmentation strategy that includes the following:

1. Reorienting angles: randomly rotating images within a specified range (−10 to 10 degrees) to create variations in lobster orientation.
2. Adjusting saturation: modifying the saturation levels in images to simulate different lighting conditions.
3. Flipping images: creating horizontal and vertical flips of the original images to introduce variations in the dataset.
4. Translating: shifting the images horizontally and vertically within a defined range to create positional variations.

With the four augmentation techniques applied to each of the 1300 original images, 5200 additional images were generated, resulting in a total dataset size of 6500 images.

As lobsters have one body, one head, one tail (folded or not), and two claws, their proportions are not equal. This uneven distribution could affect training and classification. Data augmentation techniques have helped the model to learn robust features for each class. Furthermore, the pre-trained YOLOv3, YOLOv4, and YOLOv7 models have been trained on the extensive dataset, MS COCO. These pre-trained models have already learned valuable features for detecting objects by leveraging knowledge from larger datasets.

2.3. Real-Time Object Detection

Object detection is a computer-vision task that includes classifying and locating multiple objects in a single image, using bounding boxes that locate objects in the image in order to predict the class of objects, as in image classification, and the coordinates of the bounding box which adapt to the detected object.

There are two categories of object detectors. The first category is single-stage detectors, which use a single convolutional neural network to detect objects in images, such as YOLO (You Only Look Once), a popular object detection algorithm that has seen several versions [19–23], with YOLO v7 being the latest version. It includes SSD, Single Shot Multibox Detector [24], Focal Loss for Dense Object Detection [25], and DetectoRS, Detecting Objects with Recursive Feature Pyramid and Switchable [26]. The second category includes two-stage detectors, such as Fast R-CNN [27], Faster R-CNN [28], and Mask RCNN [29]. Through a region proposal network (RPN), this category of models generates regions of interest in the first stage and then sends these region proposals to the second stage for object classification and bounding-box regression. Two-stage models are generally slower than single-stage detectors, which use a single neural network to output classification probabilities and bounding box regression.

2.4. Lobster Orientation Estimation Approach

In a controlled environment, where the workspace is well-defined with a known relationship between the camera's field of view and the robotic arm's workspace, the 2D bounding box coordinates and lobster orientation estimation provide precise information on the lobster's location and orientation, enabling a robotic arm to move quickly and accurately to pick up and manipulate the lobster for processing. This approach can lead to enhanced efficiency and increased productivity.

As illustrated in Figure 2, we developed an orientation estimation algorithm that combines a convolutional neural network model for object detection with angle calculation between the detected parts of the lobster. The algorithm proceeds through the following steps:

a. Output layers are obtained from the trained neural network model for object detection.
b. The algorithm iterates through each output layer, examining every detected object within it.
c. For each object, class scores are computed, and the class with the highest score is identified. This step helps to identify the specific part of the lobster that the bounding box corresponds to.
d. The detection confidence level is determined. If the confidence level is above a predefined threshold, the bounding box is considered reliable, and its coordinates are extracted.
e. The center point of each bounding box is calculated by averaging the x and y coordinates of the box's corners.
f. Using the center points of the bounding boxes, the angle between the line connecting the centers and the horizontal x-axis is computed using the arctan2 function, as shown in Equation (1). This angle represents the orientation of the lobster in the image.

$$\mathrm{Angle} = \mathrm{arctan2}(\mathrm{y}_2 - \mathrm{y}_1,\ \mathrm{x}_2 - \mathrm{x}_1) \tag{1}$$

The described algorithm efficiently narrows down candidate objects based on their class scores and confidence levels. This ensures that only relevant and reliable detections are considered. This filtering step minimizes false positives and maintains the performance overall. Once the bounding box coordinates are obtained, the center point is calculated and the arctan2 function is employed to determine whether the angle is computationally efficient. It provides a complete range of angles from $-\pi$ to π. While this approach has its merits, it is important to consider that the accuracy and reliability of the estimated orientation are highly dependent on the quality of the object detection model and the precision of the calculated bounding box coordinates.

In this work, we used YOLOv7. As a member of the YOLO (You Only Look Once) family, YOLOv7 is known for its real-time object detection capabilities, providing both high speed and accuracy in detecting objects within images. This advantage is crucial when working with a robotic arm, as the swift and precise localization of the lobster is required for effective manipulation. Additionally, the features of YOLOv7 are significantly improved compared to its predecessors in terms of performance, resulting in a better detection of small objects and reduced false positives. This enhanced detection quality is essential when estimating the orientation of lobsters as they may vary in size and shape. Furthermore, YOLOv7 handles a wide range of object categories, making it a versatile choice for various applications beyond lobster orientation estimation. By integrating YOLOv7 with the proposed orientation estimation algorithm, we can benefit from its real-time performance, improved accuracy, and versatility, leading to a more reliable and efficient system for guiding a robotic arm to locate and handle lobsters.

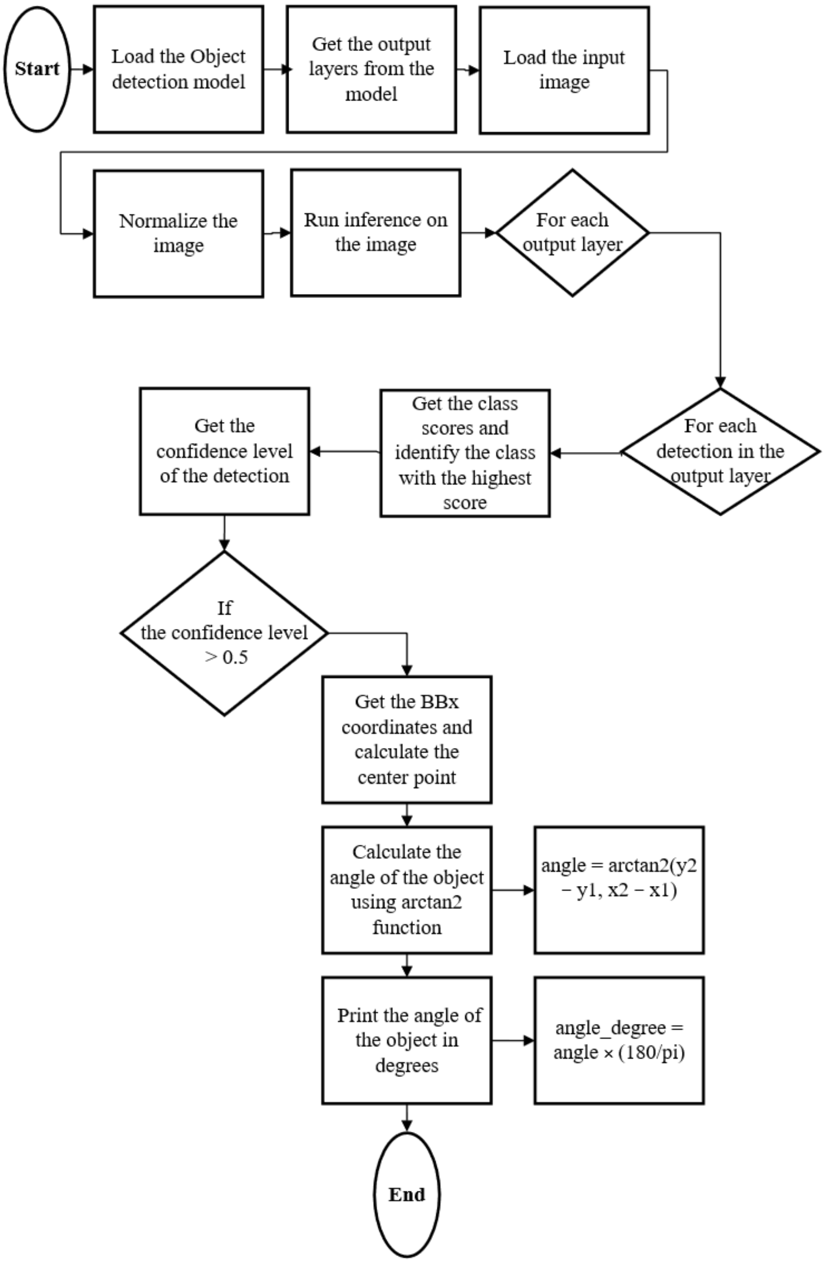

Start
Load the Object detection model
Get the output layers from the model
Load the input image
Normalize the image
Run inference on the image
For each output layer
For each detection in the output layer
Get the class scores and identify the class with the highest score
Get the confidence level of the detection
If the confidence level > 0.5
Get the BBx coordinates and calculate the center point
Calculate the angle of the object using arctan2 function
angle = arctan2(y2 − y1, x2 − x1)
Print the angle of the object in degrees
angle_degree = angle × (180/pi)
End

Figure 2. Lobster orientation estimation flowchart.

As shown in Figure 3, YOLOv7 is based on a single CNN, which is divided into three main parts: the backbone, the neck, and the head. The backbone is responsible for extracting features from the input image. In YOLOv7, the backbone is a combination of a lightweight and a deeper CNN, allowing a balance between accuracy and speed. The neck is responsible for fusing the features from the backbone, providing a higher-level representation of the input image. In YOLOv7, the neck comprises several layers of convolutional and upsampling layers. The head is responsible for predicting the bounding boxes and class probabilities of the objects in the image. In YOLOv7, the head consists of several layers of convolutional and fully connected layers. The head takes the features from the neck as an input and produces the final predictions. YOLOv7 also employs anchor boxes, which are predefined bounding boxes with various aspect ratios, to improve detection accuracy. The model uses a prediction module that estimates class probabilities and bounding box coordinates for each anchor box.

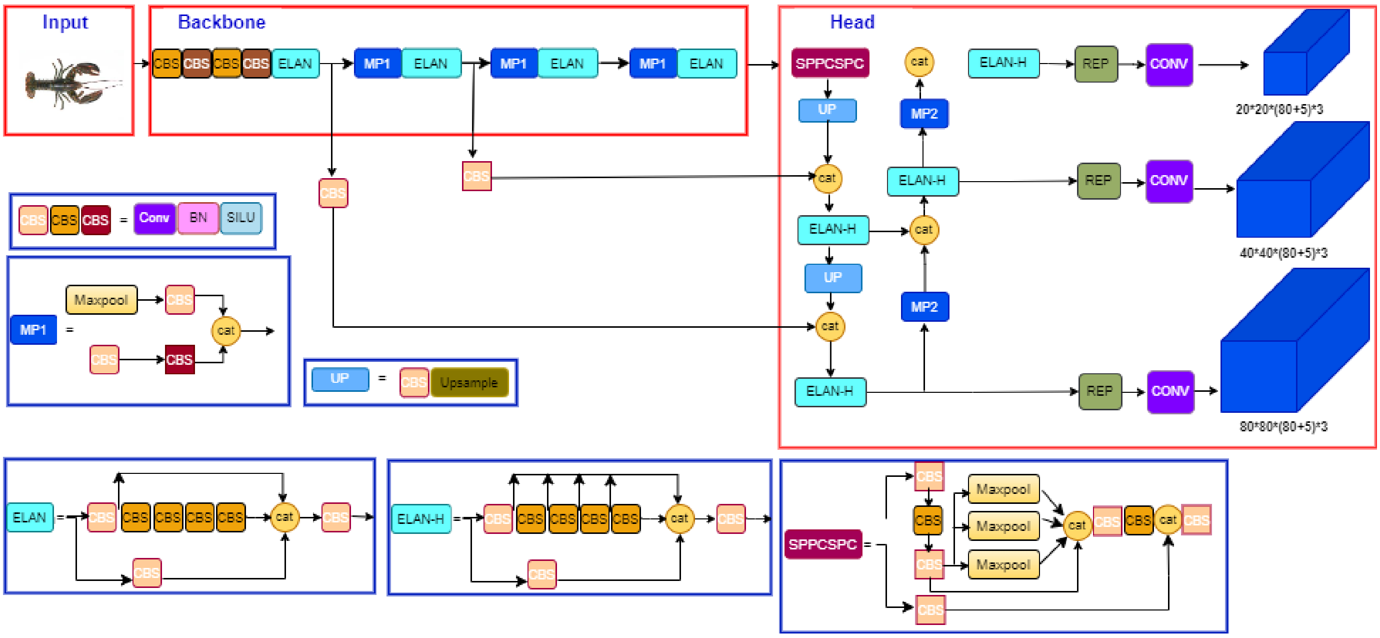


Figure 3. YOLOv7 architecture.

2.5. Experimental Setup

Figure 4 shows the experimental environment for this study. The training was conducted on the Ubuntu operating system using an Acer model computer equipped with Intel Core i7-8750H @ 2.20 GHz, GPU Nvidia GeForce RTX 3060 Ti and 16 GB RAM. Then, the models were deployed and experimented on using the embedded mobile platform, Nvidia Jetson Xavier NX, which provides high artificial intelligence performance, the type of power efficiency needed for all modern AI networks, and the platform is supported by Nvidia software Jet-Pack SDK, which includes CUDA Toolkit, cuDNN, OpenCV, TensorRT, and L4T with the LTS Linux Kernel. Table 1 shows the hardware specifications of Jetson Xavier NX.

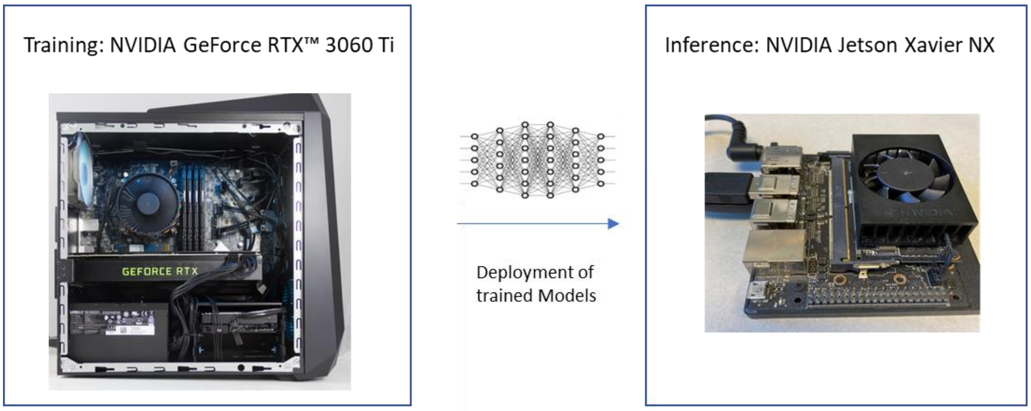

Training: NVIDIA GeForce RTX™ 3060 Ti
GEFORCE RTX
Deployment of
trained Models
Inference: NVIDIA Jetson Xavier NX

Figure 4. Experimental material.

Table 1. Nvidia Jetson NX hardware specification.

Embedded Environment	Jetson Xavier NX
CPU	6-core NVIDIA Carmel ARM®v8.2 64-bit CPU 6 MB L2 + 4 MB L3
GPU	NVIDIA Volta architecture with 384 NVIDIA CUDA® cores and 48 Tensor cores
Memory	8 GB
Storage	microSD 128 GB
size	70 mm × 45 mm

Figure 5 illustrates the steps involved in implementing object detection models for detecting American lobster parts. Initially, images were manually annotated using LabelImg, an open-source graphical image annotation tool, as demonstrated in Figure 6. The labeling results were saved directly in YOLO format, with a text file accompanying each image, sharing the same name as its corresponding image file. Each line within the text file represents the attributes of a single object (class number, object center in x, object center in y, object width, and object height). Subsequently, the image files were partitioned into two sets: 90% for training and 10% for testing. All text files and image sets were then input into the training process, where transfer learning was employed, and model hyperparameters were fine-tuned. The size of the input images was 640 × 640. The YOLOv7, YOLOv7-Tiny, YOLOv4, and YOLOv3 algorithms, which were previously trained on the MS COCO dataset [30], were retrained using GPU. Following the training process, performance metrics and visual detection were evaluated to select the best-performing weights. For further testing and evaluation, the trained models were implemented on the Nvidia Jetson Xavier NX platform.

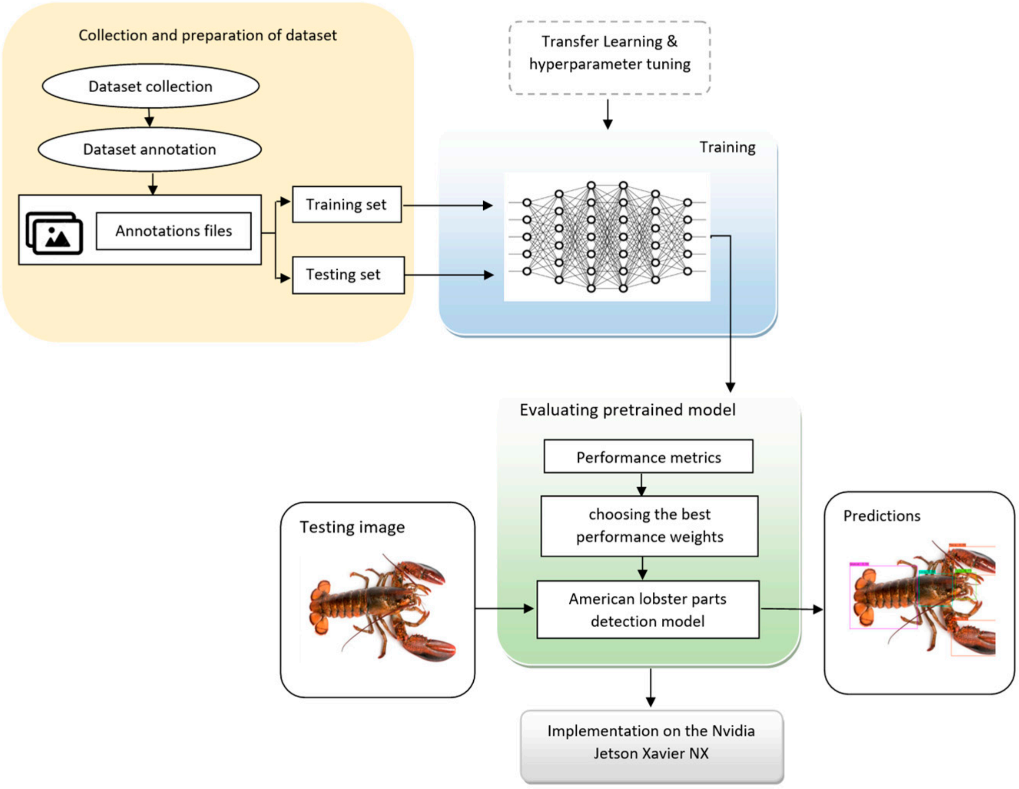
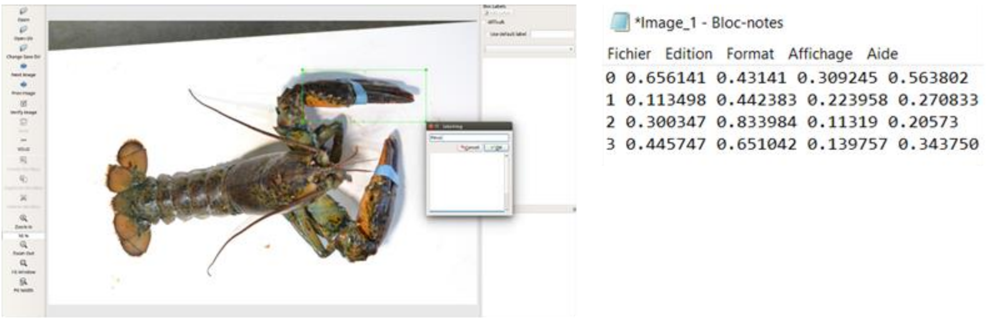

Collection and preparation of dataset
Dataset collection
Dataset annotation
Annotations files
Training set
Testing set
Transfer Learning & hyperparameter tuning
Training
Evaluating pretrained model
Performance metrics
choosing the best performance weights
American lobster parts detection model
Testing image
Predictions
Implementation on the Nvidia Jetson Xavier NX

Figure 5. Methodology of Implementation.



```
0 0.656141 0.43141 0.309245 0.563802
1 0.113498 0.442383 0.223958 0.270833
2 0.300347 0.833984 0.11319 0.20573
3 0.445747 0.651042 0.139757 0.343750
```

Figure 6. Graphical image annotation tool LabelImg 1.7.0.

2.6. Model Performance Metrics

The performance of the models was assessed using standard performance metrics commonly employed in object detection tasks [31]. These metrics are crucial for comparing different models and determining their effectiveness at detecting objects accurately. The following section defines some basic concepts used in the calculation of these performance metrics:

- TP (true positive) is the number of objects detected correctly.
- FP (false positive) is the number of objects incorrectly detected as positive.
- FN (false negative) is the number of objects incorrectly detected as negative (missed ground truth).

To compute these values, the Generalized Intersection Over Union (GIOU) score is used in order to determine if the detection is correct or not by comparing the GIOU score to a predefined threshold. The GIOU measures how well the predicted bounding box overlaps the ground truth bounding by taking into account the differences in the size and aspect ratio between the predicted and ground truth boxes, in addition to their overlap area and union area, as shown in Figure 7. GIOU has shown itself to be more robust than Intersection Over Union (IOU), especially when dealing with small or heavily overlapping objects.

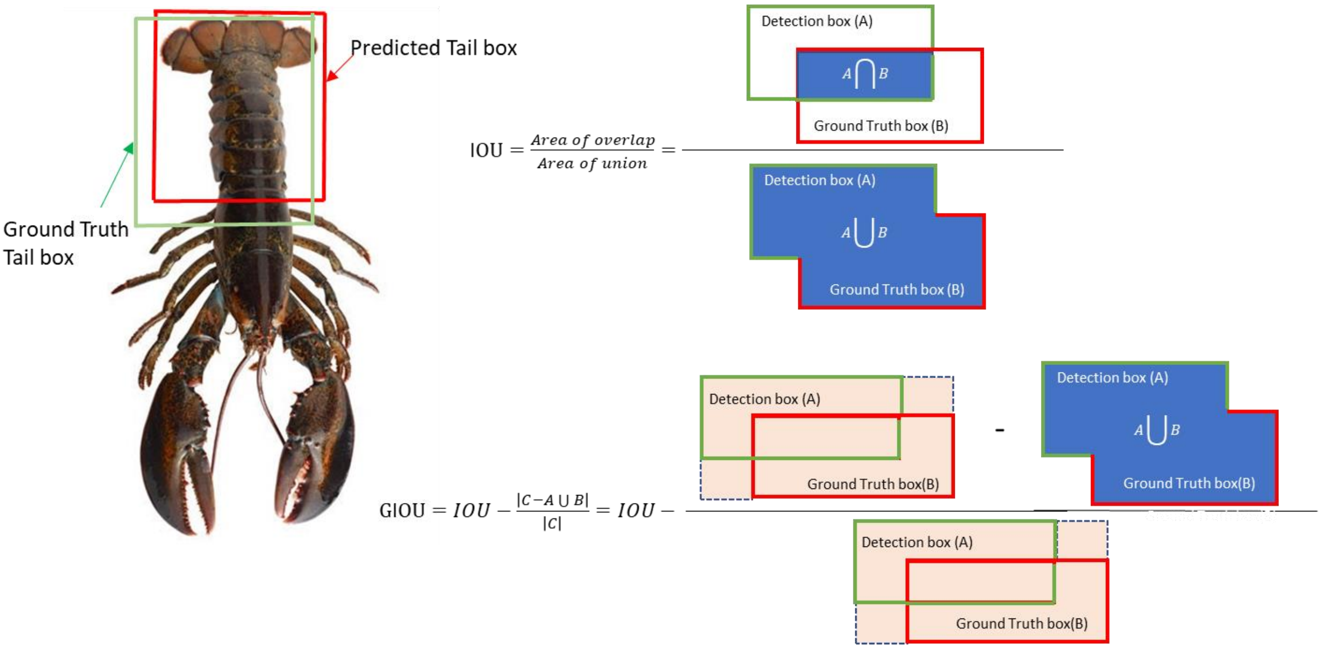


Figure 7. Generalized Intersection Over Union (GIOU).

IOU is the Intersection Over Union score, A is the area of the predicted bounding box, B is the area of the ground-truth bounding box, C is the area of the smallest box that completely encloses both A and B, and (A U B) is the area of the union of A and B.

The GIOU score ranges from −1 to 1, where 1 indicates a perfect match between the predicted and ground-truth bounding boxes, 0 indicates no match, and −1 indicates a complete mismatch.

Once the GIOU score is calculated for all images, precision, recall, and F1-score metrics can be calculated. Precision, which measures the accuracy of the model to identify a sample as positive, is computed with the following equation:

$$\text{Precision} = \left(\frac{\text{TP}}{\text{TP} + \text{FP}}\right) \tag{2}$$

Recall, which measures the ability of the model to identify all the positive samples as positive, is computed with the following equation:

$$\text{Recall} = \left(\frac{\text{TP}}{\text{TP} + \text{FN}}\right) \tag{3}$$

High precision means a low false-positive prediction rate and high recall means a low false-negative prediction rate. Hence, an accurate object detection model should keep a balance between precision and recall, at a fixed recall interval [0 1.0], with steps of 0.1, according to the 11-point interpolation method proposed by Gerard Salton [32]. The precision and recall curve is summed with the Average Precision (AP) metric and computed with the following equation:

$$\text{AP}_{11} = \frac{1}{11} \sum_{\text{R}\epsilon\{0,0.1,\ldots,0.9,1\}}^{\text{n}} \text{P}_{\text{interp}}(\text{R}) \tag{4}$$

where

$$\text{P}_{\text{interp}}(\text{R}) = \max_{\text{R}',\text{R}' \geq \text{R}} \text{P}(\text{R}') \tag{5}$$

This means that rather than using observed precision at each point R, the AP is calculated by taking the maximum precision at a recall that is greater than or equal to R.

F1 score is the harmonic mean of precision and recall; it maintains the balance between precision and recall [33], and is computed with the following equation:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

The mean Average Precision (mAP) metric measures the object detector's accuracy over all specific classes. In other words, the mAP is the average AP over all classes [33], and is computed with the following equation:

$$\text{mAP} = \frac{\sum_{\text{i}=1}^{\text{N}} \text{AP}_{\text{i}}}{\text{N}} \tag{7}$$

where AP_i represents the AP of the class i and N is the number of all evaluated classes.

Frames per second (fps) represents the number of images that can be detected per second and provides an evaluation of the detector speed.

The precision, recall, F1 score, and AP were calculated independently for each class, treating each class as a positive class and the remaining classes as negatives. The overall performance of the model was then calculated by averaging these metrics across all classes.

3. Results and Discussion

An experimental evaluation of the YOLOv7 model was conducted on the lobster dataset. The results of the model are illustrated in Figure 8, which shows a high degree of detection efficiency for the various target classes. A value of 96.2% was calculated for the model's mean average precision (mAP). As a result of this high level of precision, YOLOv7 is capable of identifying and distinguishing lobster body parts with high accuracy. For accurate orientation estimation and manipulation, this is crucial.

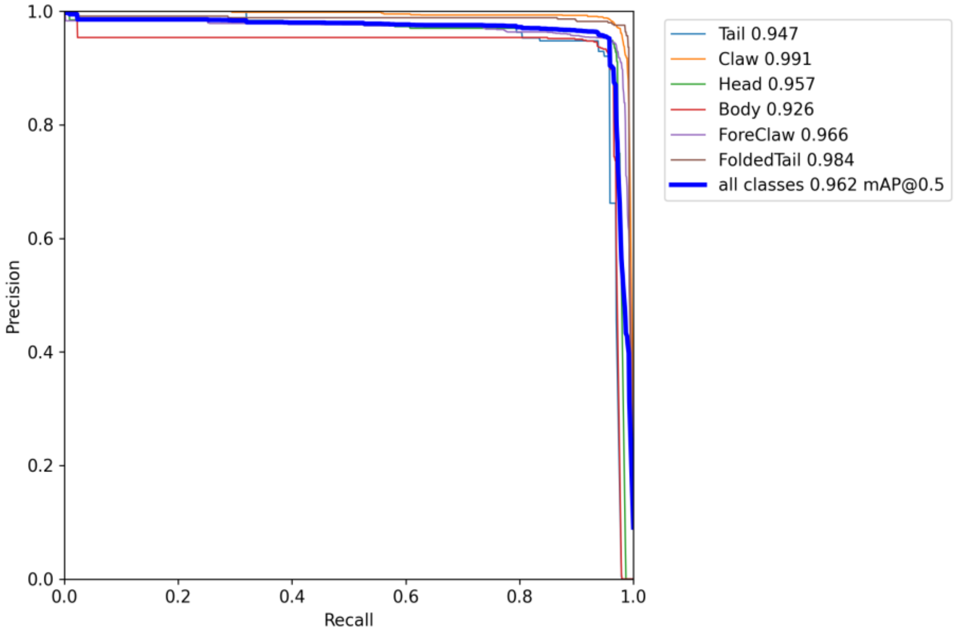


Figure 8. The precision–recall curve of the YOLOv7 model on the lobster dataset.

The YOLOv7 model's accuracy was assessed using a confusion matrix as a performance metric. As depicted in Figure 9, each column represents the predicted proportions for each class. Each row corresponds to the actual proportions of each class present in the data. As per the data presented in Figure 9, the model demonstrates a high accuracy in predicting the classes "Tail", "Claw", "Head", "Body", "Fore-Claw", and "Folded-Tail", with correct prediction rates of 97%, 98%, 97%, 96%, 96%, and 97%, respectively. This evidence of the model's accuracy highlights its effectiveness in classifying and identifying various lobster parts. This suggests that YOLOv7 is well-suited to applications requiring precise detection and distinction of complex object classes.

As part of this study, the changes in the loss values, including the Box loss, the objectness loss, and the classification loss are presented in a graphical format. YOLOv7 uses the 'GIOU Loss' as the bounding box loss function. Box loss is calculated as the mean of the GIOU loss. A higher accuracy is indicated by a lower box loss value. Objectness loss measures the difference between the predicted and ground truth objectness scores, with a lower value indicating a higher accuracy. The classification loss measures the difference between the predicted and ground truth class probabilities for each object, where a lower value represents a higher accuracy. As shown in Figure 10, as iterations increase, loss values steadily decrease and eventually stabilize; after 200 iterations, convergence is achieved.

Additionally, YOLOv7 was benchmarked against other well-known object detection models, including YOLOv3, YOLOv4, and YOLOv7-Tiny, to demonstrate its effectiveness in detecting lobster body parts. Training and testing were conducted using the lobster dataset. As part of the evaluation process, performance metrics including precision, recall, F1-score, and mAP@0.5 were evaluated. Based on the performance metrics presented in Table 2, it is evident that YOLOv7 outperforms the other object detection models by a considerable margin. According to the results, this model achieved scores of 95.5%, 95.1%, 95.2%, and 95.3%, respectively, for precision, recall, F1-score, and mAP@0.5, demonstrating its superior capability when compared with other models in detecting and identifying lobster body parts. In this study, the YOLOv7 object detection model was found to be the top performer and was therefore chosen to be used in further experiments.

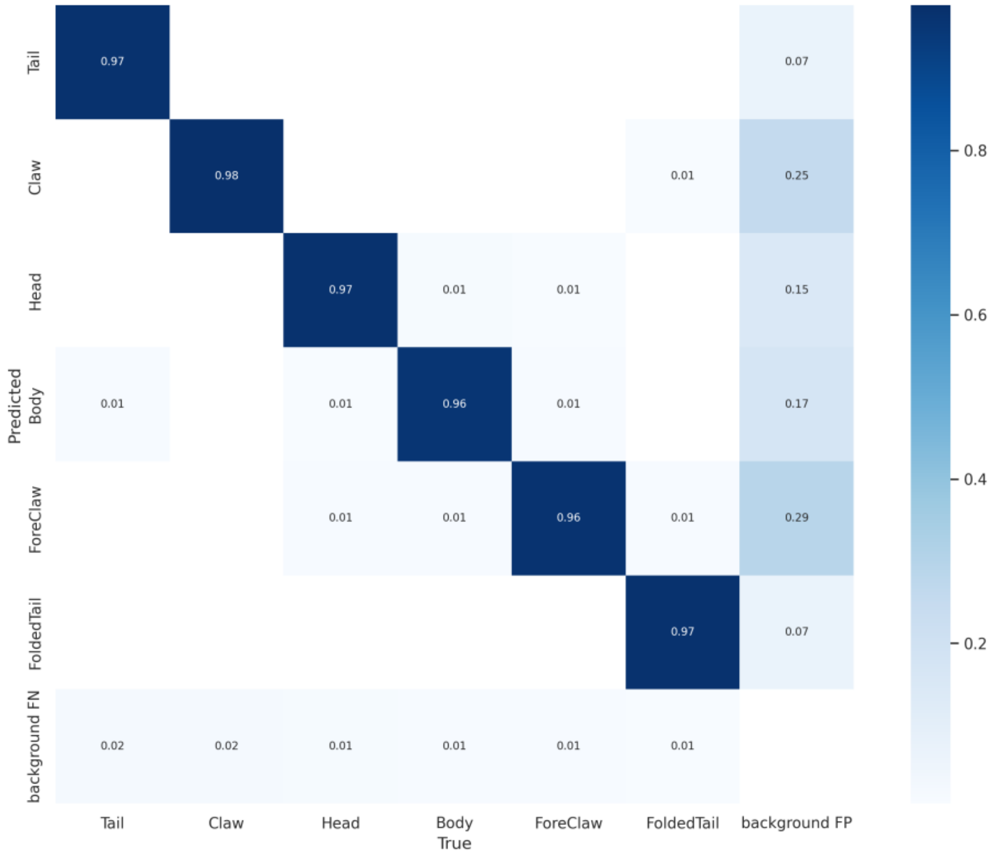


Figure 9. The confusion matrix of the YOLOv7 model on the lobster dataset.

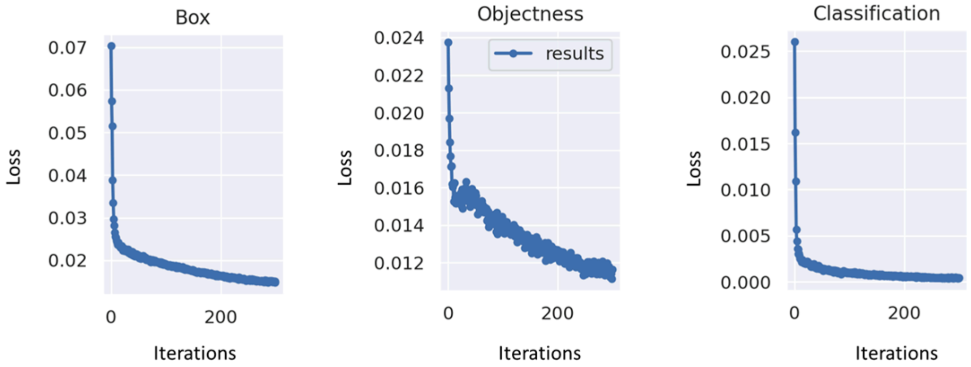


Figure 10. Loss values on the lobster dataset: variation curves.

Table 2. Result of Training.

Detector	Precision (%)	Recall (%)	F1 Score (%)	mAP@0.50 (%)
YOLOv3	87	89	88	83.5
YOLOv4	94	95	94.5	91.8
YOLOv7	95.5	95.1	95.2	95.3
YOLOv7-Tiny	84.4	78.9	81.5	80

On the GeForce RTX 3060 Ti, as shown in Table 3, the YOLOv7 detector was able to achieve an impressive detection speed of 111 frames per second. This demonstrates its ability to detect objects at high speeds. This speed is slightly slower than YOLOv7-Tiny's 188.7 fps, but it is important to note that YOLOv7 still delivers a remarkable performance in terms of frame rates. The model is able to handle real-time applications effectively, even when compared to its faster counterpart, the YOLOv7-Tiny.

Table 3. Performance comparison of YOLOv3, YOLOv4, YOLOv7, and YOLOv7-tiny detectors on the Geforce RTX 3060 Ti and the Jetson Xavier NX.

Detector	Nvidia Platform	Inference Time (ms)	FPS
YOLOv3	Geforce RTX 3060 Ti	12.62	79
	Jetson Xavier NX	179.46	5.6
YOLOv4	Geforce RTX 3060 Ti	13.33	75
	Jetson Xavier NX	180.54	5.5
YOLOv7	Geforce RTX 3060 Ti	9	111
	Jetson Xavier NX	116.73	8.6
YOLOv7-Tiny	Geforce RTX 3060 Ti	5.3	188.7
	Jetson Xavier NX	39	25.6

Models were deployed on the NVIDIA Jetson Xavier NX embedded platform. A comparison of the trained models' inference time detection is presented in Table 3. To facilitate real-time evaluation, the inference time has been converted from milliseconds to frames per second. A powerful device, the NVIDIA Jetson Xavier NX, allowed YOLOv7-Tiny to achieve 25.6 frames per second in real-time. However, YOLOv7 was deemed unsuitable for deployment on mobile devices due to its high computational requirements. It can be seen from Table 3 that YOLOv7 was only able to achieve an average frame rate of 8.6, making it infeasible for real-time detectors to operate on the Jetson Xavier NX. In contrast, the real-time performance of YOLOv7-Tiny on the Jetson Xavier NX is quite promising as it achieved a frame rate of 25.6 frames per second. This indicates that YOLOv7-Tiny may be a more suitable choice for real-time applications on resource-constrained platforms such as the Jetson Xavier NX.

Figure 11 demonstrates the performance of the Jetson Xavier NX device in terms of both its accuracy (mAP@0.5) and inference time (fps). The evaluation results clearly indicate that YOLOv7-Tiny emerges with the highest score of 105.6. This is followed closely by YOLOv7 and YOLOv4, which score 103.9 and 97.3 points, respectively. This comparison emphasizes the balance between detection accuracy and speed.

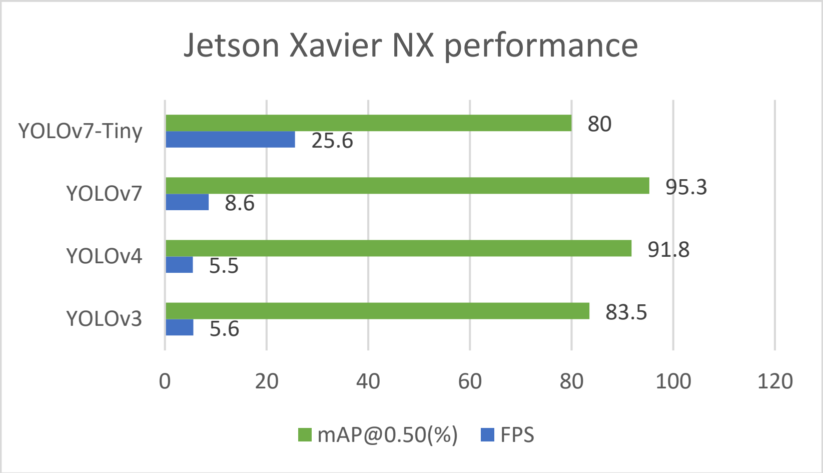
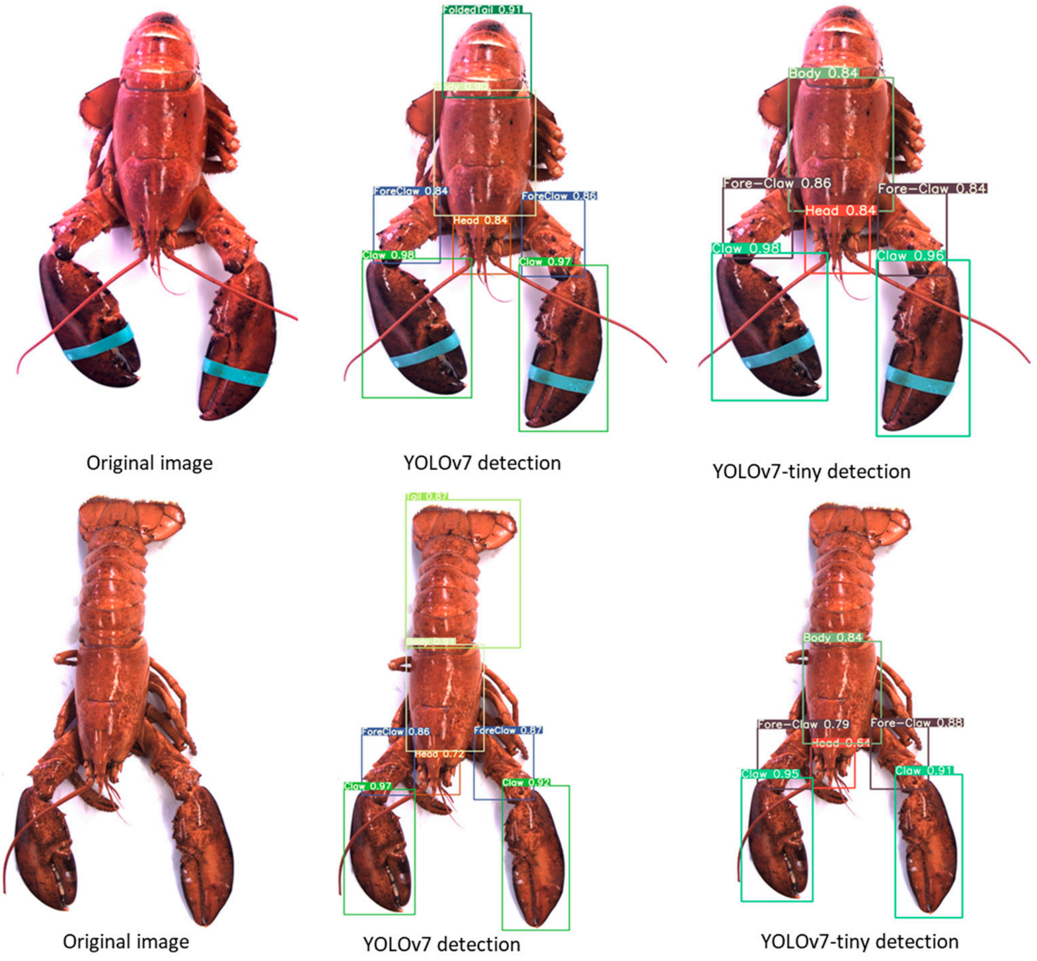


Figure 11. Jetson Xavier NX performance of 4 pre-trained YOLO models.

In an industrial context, with a distance between the camera and the processing line of about 50–100 cm, and the processing line normally running at about 30 m per second (m/s), the vision system should have a speed of 30 to 60 fps [33]. According to the results of the experiments, the Jetson Xavier NX can achieve real-time performance (25.6 fps) with YOLOv7-Tiny, but not with YOLOv7 (8.6 fps), which requires a high computing device for real-time lobster processing applications.

The visual detection results illustrated in Figure 12 for both YOLOv7 and YOLOv7-tiny algorithms that were trained on American lobster images demonstrate that YOLOv7 was highly successful in identifying all lobster body parts with high scores, showcasing the effectiveness of the full model for this specific task. On the other hand, YOLOv7-tiny, a smaller and more compact version of the model, failed to detect some parts of the lobster. This discrepancy can be attributed to the reduced complexity and computational capacity of the YOLOv7-tiny model, which sacrifices some accuracy for the sake of increased speed and reduced resource consumption.

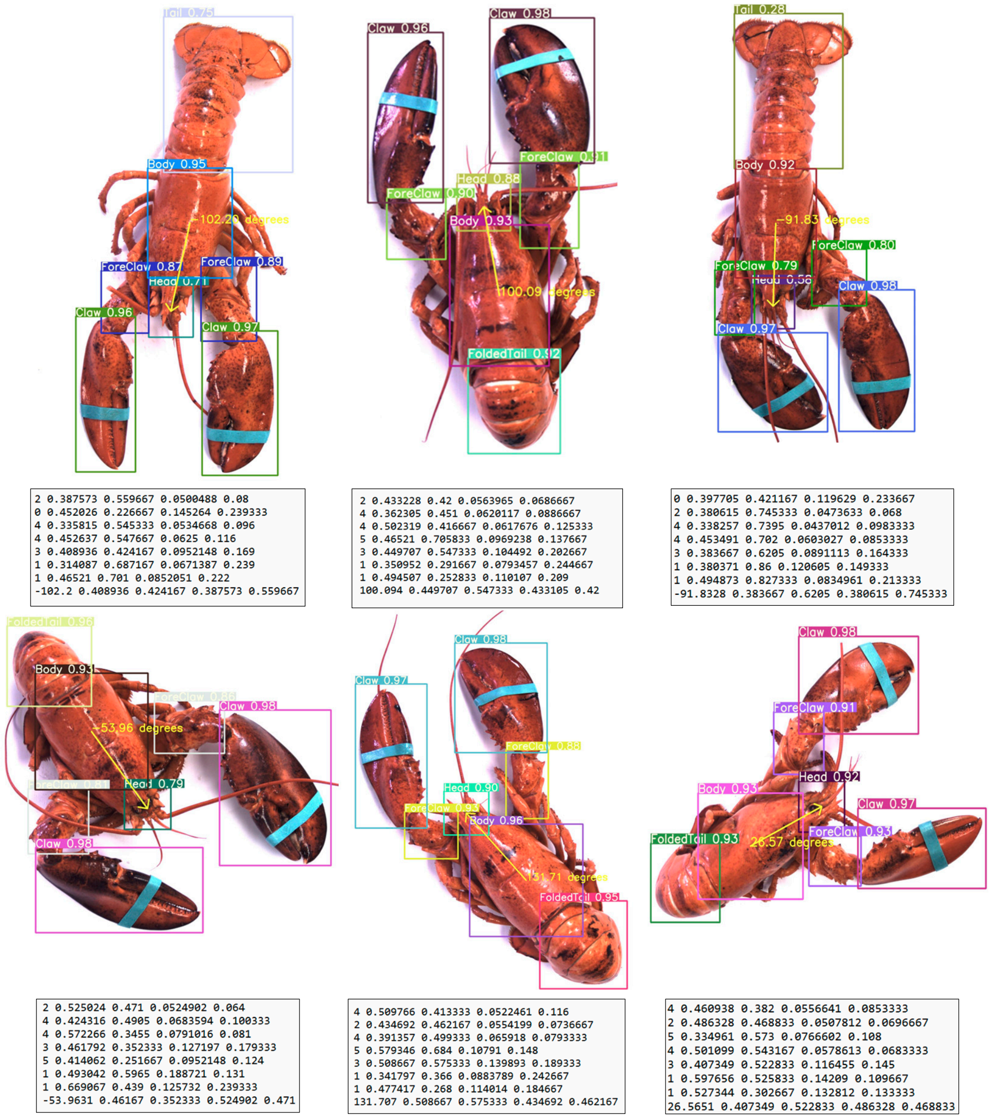


Original image YOLOv7 detection YOLOv7-tiny detection

Original image YOLOv7 detection YOLOv7-tiny detection

Figure 12. American lobster parts visual detection results based on YOLOv7 and YOLOv7-tiny algorithms.

YOLOv7 was ultimately chosen for estimating lobster orientation, leveraging its proven ability to accurately detect lobster body parts. This approach utilized the center coordinates of the head and body bounding boxes to estimate the orientation. By calculating the relative positions and angles between these two points, the model was able to infer the overall direction in which the lobster was facing. The successful implementation of YOLOv7 in this task can be seen in the results, with the lobster orientation estimation and corresponding output vectors clearly illustrated in Figure 13. These findings demonstrate the efficacy of YOLOv7 in not only detecting lobster body parts but also in extracting valuable information about their spatial orientation. In addition, this model's capability to process and analyze lobster structural details highlights its adaptability to handle similar challenges in other species or objects with complex morphologies.



```
2 0.387573 0.559667 0.0500488 0.08
0 0.452026 0.226667 0.145264 0.239333
4 0.335815 0.545333 0.0534668 0.096
4 0.452637 0.547667 0.0625 0.116
3 0.408936 0.424167 0.0952148 0.169
1 0.314087 0.687167 0.0671387 0.239
1 0.46521 0.701 0.0852051 0.222
-102.2 0.408936 0.424167 0.387573 0.559667
```

```
2 0.433228 0.42 0.0563965 0.0686667
4 0.362305 0.451 0.0620117 0.0886667
4 0.502319 0.416667 0.0617676 0.125333
5 0.46521 0.705833 0.0969238 0.137667
3 0.449707 0.547333 0.104492 0.202667
1 0.350952 0.291667 0.0793457 0.244667
1 0.494507 0.252833 0.110107 0.209
100.094 0.449707 0.547333 0.433105 0.42
```

```
0 0.397705 0.421167 0.119629 0.233667
2 0.380615 0.745333 0.0473633 0.068
4 0.338257 0.7395 0.0437012 0.0983333
4 0.453491 0.702 0.0603027 0.0853333
3 0.383667 0.6205 0.0891113 0.164333
1 0.380371 0.86 0.120605 0.149333
1 0.494873 0.827333 0.0834961 0.213333
-91.8328 0.383667 0.6205 0.380615 0.745333
```

```
2 0.525024 0.471 0.0524902 0.064
4 0.424316 0.4905 0.0683594 0.100333
4 0.572266 0.3455 0.0791016 0.081
3 0.461792 0.352333 0.127197 0.179333
5 0.414062 0.251667 0.0952148 0.124
1 0.493042 0.5965 0.188721 0.131
1 0.669067 0.439 0.125732 0.239333
-53.9631 0.46167 0.352333 0.524902 0.471
```

```
4 0.509766 0.413333 0.0522461 0.116
2 0.434692 0.462167 0.0554199 0.0736667
4 0.391357 0.499333 0.065918 0.0793333
5 0.579346 0.684 0.10791 0.148
3 0.508667 0.575333 0.139893 0.189333
1 0.341797 0.366 0.0883789 0.242667
1 0.477417 0.268 0.114014 0.184667
131.707 0.508667 0.575333 0.434692 0.462167
```

```
4 0.460938 0.382 0.0556641 0.0853333
2 0.486328 0.468833 0.0507812 0.0696667
5 0.334961 0.573 0.0766602 0.108
4 0.501099 0.543167 0.0578613 0.0683333
3 0.407349 0.522833 0.116455 0.145
1 0.597656 0.525833 0.14209 0.109667
1 0.527344 0.302667 0.132812 0.133333
26.5651 0.407349 0.522833 0.486328 0.468833
```

Figure 13. American Lobster orientation estimation.

4. Conclusions

This study demonstrated the integration of an orientation estimation algorithm with a convolutional neural network model, specifically YOLOv7, to estimate lobster orientation in images. Through rigorous comparison with other models, including YOLOv7-tiny, YOLOv4, and YOLOv3, YOLOv7 emerged as the top performer in terms of accuracy and inference time, boasting a mean average precision (mAP) of 95.3 and 111 FPS on the GeForce RTX 3060 Ti. However, when deployed on the NVIDIA Jetson Xavier NX, YOLOv7′s performance dropped to 8 FPS, rendering it unsuitable for real-time applications on this platform. Nevertheless, the study adopted YOLOv7 for lobster orientation estimation due to its superior performance, with the aim of guiding the FANUC LR Mate 200 iD robotic arm in lobster manipulation tasks within the robot's workspace. This novel approach has the potential to overcome the limitations of FANUC's IRVision system, which previously struggled to detect complex lobster body parts, and paves the way for more efficient and accurate lobster processing in the food industry.

Future work should aim to explore several areas to build upon these findings, including experimenting with the FANUC LR Mate 200 iD robotic arm for lobster manipulation using a YOLOv7-based orientation estimation. Optimizing YOLOv7 for deployment on platforms with limited computational resources, such as the NVIDIA Jetson Xavier NX, will help towards achieving high-quality real-time performance without sacrificing accuracy.

Author Contributions: Conceptualization, N.C. and Y.B.; methodology, N.C. and Y.B.; software, N.C.; validation, N.C., Y.B. and M.G.; formal analysis, N.C.; investigation N.C.; resources, N.C. and Y.B.; data curation, N.C. and Y.B.; writing—original draft preparation, N.C.; writing—review and editing, N.C. and Y.B.; supervision, Y.B. and M.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors express their thanks to the New Brunswick Innovation Foundation (NBIF) and the Atlantic Canada Opportunities Agency Canada (ACOA) for the experimental equipment.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Renaud, I.; Yacout, S. Decision support system for quality assurance programs in the fish and seafood processing industry. *Comput. Ind. Eng.* **1995**, *29*, 31–35. [CrossRef]
2. An Exploration of Skills and Labour Shortages in Atlantic Canada—Canada.ca. Available online: https://www.canada.ca/en/atlantic-canada-opportunities/services/researchstudies2.html (accessed on 13 September 2021).
3. MacEachern, C.B.; Esau, T.J.; Schumann, A.W.; Hennessy, P.J.; Zaman, Q.U. Detection of fruit maturity stage and yield estimation in wild blueberry using deep learning convolutional neural networks. *Smart Agric. Technol.* **2023**, *3*, 100099. [CrossRef]
4. Zhou, L.; Zhang, C.; Liu, F.; Qiu, Z.; He, Y. Application of Deep Learning in Food: A Review. *Compr. Rev. Food Sci. Food Saf.* **2019**, *18*, 1793–1811. [CrossRef] [PubMed]
5. Zhu, H.; Deng, L.; Wang, D.; Gao, J.; Ni, J.; Han, Z. Identifying carrot appearance quality by transfer learning. *J. Food Process. Eng.* **2019**, *42*, 1–10. [CrossRef]
6. Xu, J.; Liong, S.; Tan, L.; Gan, Y.S. Find the centroid: A vision-based approach for optimal object grasping. *J. Food Process. Eng.* **2021**, *44*, 1–18. [CrossRef]
7. Ji, W.; Gao, X.; Xu, B.; Pan, Y.; Zhang, Z.; Zhao, D. Apple target recognition method in complex environment based on improved YOLOv4. *J. Food Process. Eng.* **2021**, *44*, e13866. [CrossRef]
8. Zhao, S.; Hao, G.; Zhang, Y.; Wang, S. A real-time classification and detection method for mutton parts based on single shot multi-box detector. *J. Food Process. Eng.* **2021**, *44*, e13749. [CrossRef]
9. Dairath, M.H.; Mehmood, M.A.; Sarwar, H.U.; Akram, M.Z.; Omar, M.M.; Faheem, M. Computer vision-based prototype robotic picking cum grading system for fruits. *Smart Agric. Technol.* **2023**, *4*, 100210. [CrossRef]
10. Liu, Z.; Jia, X.; Xu, X. Study of shrimp recognition methods using smart networks. *Comput. Electron. Agric.* **2019**, *165*, 104926. [CrossRef]
11. Fernandes, A.F.; Turra, E.; de Alvarenga, R.; Passafaro, T.L.; Lopes, F.B.; Alves, G.F.; Singh, V.; Rosa, G.J. Deep Learning image segmentation for extraction of fish body measurements and prediction of body weight and carcass traits in Nile tilapia. *Comput. Electron. Agric.* **2020**, *170*, 105274. [CrossRef]
12. Xie, T.; Li, X.; Zhang, X.; Hu, J.; Fang, Y. Detection of Atlantic salmon bone residues using machine vision technology. *Food Control.* **2021**, *123*, 107787. [CrossRef]
13. Liu, H.-W.; Chen, C.-H.; Tsai, Y.-C.; Hsieh, K.-W.; Lin, H.-T. Identifying Images of Dead Chickens with a Chicken Removal System Integrated with a Deep Learning Algorithm. *Sensors* **2021**, *21*, 3579. [CrossRef]
14. Zhang, L.; Gray, H.; Ye, X.; Collins, L.; Allinson, N. Automatic Individual Pig Detection and Tracking in Pig Farms. *Sensors* **2019**, *19*, 1188. [CrossRef]
15. Salau, J.; Krieter, J. Instance Segmentation with Mask R-CNN Applied to Loose-Housed Dairy Cows in a Multi-Camera Setting. *Animals* **2020**, *10*, 2402. [CrossRef]
16. Zheng, Y.-Y.; Kong, J.-L.; Jin, X.-B.; Wang, X.-Y.; Su, T.-L.; Zuo, M. CropDeep: The Crop Vision Dataset for Deep-Learning-Based Classification and Detection in Precision Agriculture. *Sensors* **2019**, *19*, 1058. [CrossRef]
17. This Robot Can Process a Crab in Seconds, and It Might Actually Save Rural Jobs. CBC News. Available online: https://www.cbc.ca/news/canada/newfoundland-labrador/robotic-processing-seafood-1.4185530 (accessed on 10 September 2021).
18. Chelouati, N.; Fares, F.; Bouslimani, Y.; Ghribi, M. Lobster detection using an Embedded 2D Vision System with a FANUC industrual robot. In Proceedings of the 2021 IEEE International Symposium on Robotic and Sensors Environments (ROSE), FL, USA, 28–29 October 2021. [CrossRef]
19. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
20. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525. [CrossRef]
21. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.

22. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-J.M. YOLOv4 Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
23. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
24. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016. [CrossRef]
25. Qiao, S.; Chen, L.-C.; Yuille, A. DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2021, Nashville, TN, USA, 19–25 June 2021; pp. 10208–10219. [CrossRef]
26. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
27. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
28. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [CrossRef]
29. Lin, T.Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context. *arXiv* **2015**, arXiv:1405.0312.
30. Padilla, R.; Netto, S.L.; da Silva, E.A.B. A Survey on Performance Metrics for Object-Detection Algorithms. In Proceedings of the 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niteroi, Brazil, 1–3 July 2020; pp. 237–242.
31. Salton, G. *Introduction to Modern Information Retrieval*; Mcgraw Hill Computer Science Series; McGraw-Hill Book Co.: New York, NY, USA, 1983.
32. Goutte, C.; Gaussier, E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In Proceedings of the European Conference on Information Retrieval, Santiago de Compostela, Spain, 21–23 March 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 345–359.
33. Li, J.; Su, Z.; Geng, J.; Yin, Y. Real-time Detection of Steel Strip Surface Defects Based on Improved YOLO Detection Network. *IFAC-PapersOnLine* **2018**, *51*, 76–81. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.