



A convolutional neural network-based lightweight motion deblurring method for autonomous visual target tracking in bionic robotic fish

Yang Liu ^{a,b,c,d}, Bingxiong Wang ^{a,b,c,d}, Runtong Ai ^{a,b,c,d}, Guohua Yu ^{a,b,c,d}, Yinjie Ren ^{a,b,c,d}, Jincun Liu ^{a,b,c,d,*}, Yaoguang Wei ^{a,b,c,d}, Dong An ^{a,b,c,d}, Yan Meng ^{a,b,c,d}

^a National Innovation Center for Digital Fishery, China Agricultural University, Beijing, 100083, PR China

^b Key Laboratory of Smart Farming Technologies for Aquatic Animals and Livestock, Ministry of Agriculture and Rural Affairs, China Agricultural University, Beijing, 100083, PR China

^c Beijing Engineering and Technology Research Centre for Internet of Things in Agriculture, Beijing, 100083, PR China

^d College of Information and Electrical Engineering, China Agricultural University, Beijing, 100083, PR China

ARTICLE INFO

Keywords:

Image motion deblurring
Lightweight network
Visual target tracking
Bionic robotic fish

ABSTRACT

The fish-like swimming motion of robotic fish induces periodic head oscillations, which lead to motion blur in the captured images, negatively impacting autonomous visual target tracking tasks. To enhance the quality of images acquired during robotic fish tracking, this study introduces a real-time, Lightweight Motion Deblurring Network named LMDNet. The proposed network is built on a U-Net (U-shaped Network) architecture integrated with the NAF-Block (Nonlinear Activation Free Block) module from NAFNet (Nonlinear Activation Free Network). To further address feature distribution issues, the Test-time Local Converter (TLC) method is applied for global feature aggregation, and a ghost module is incorporated to generate supplementary feature information at reduced computational cost. Additionally, a wavelet loss function tailored for image motion deblurring is proposed to effectively leverage frequency features for model learning. Experimental results demonstrate that LMDNet, as a lightweight model with only 2.7 Mega (M) parameters and a computational cost of 14.6 Giga Multiply-Accumulate Operations (GMACs), achieves superior performance compared to other lightweight motion deblurring algorithms. Compared to state-of-the-art motion deblurring algorithms, LMDNet significantly reduces computational costs while maintaining good deblurring performance, with the computational complexity only 1/42nd that of Fast Fourier Transform Transformer (FFTformer). After LMDNet deblurring, the detection success rate of YOLOv5 (You Only Look Once version5) increased by 17.05 % and 11.53 %, respectively. Furthermore, this study jointly deployed the LMDNet motion deblurring and YOLOv5 target detection algorithms on the edge computing devices-equipped tuna robotic fish, enabling effective autonomous visual tracking of a small robotic shark and validating the proposed algorithm's effectiveness.

1. Introduction

Underwater vehicles play a crucial role in exploring marine environments and resources, advancing human understanding and exploration of the oceans (Liu, Anderlini, Wang, Ma, & Ding, 2022b). As a result, to develop underwater vehicles with higher swimming efficiency, environmental adaptability, and maneuverability, a growing number of researchers have drawn inspiration from marine organisms, resulting in the creation of various types of bionic robotic fish (Chen, Wu, Meng, Tan, & Yu, 2022a; Liu et al., 2021; Liu, Liu, Yang, Liu, & Liu, 2023; Meng, Wu, Dong, Wang, & Yu, 2020). Bionic robotic fish, designed by mimicking the morphology and swimming patterns of various fish species, offer superior performance in terms of maneuverability, stealth, and ef-

ficiency. Consequently, bionic robotic fish have a vast range of potential applications and are suited for activities like biological observation, surveillance, patrolling, and archaeology, which would be extremely dangerous or unfeasible with humans or traditional underwater vehicles (Cui, Li, Wang, Zhong, & Li, 2023).

Currently, the majority of research on bionic robotic fish concentrates on mechanical structures, driving mode, propulsion mechanisms, motion control and other aspects (Chen, Yang, Xu, Lu, & Hu, 2022b; Liu et al., 2022a; Qiu, Wu, Wang, Tan, & Yu, 2022; Wang et al., 2022b), while studies related to their environmental perception are relatively limited. The environmental perception capability of bionic robotic fish is mostly achieved by integrating optical and acoustic sensors (Qiu, Wu, Wang, Tan, & Yu, 2023; Sun et al.,

* Corresponding author.

E-mail addresses: liuyangcau@cau.edu.cn (Y. Liu), wangbingxiong@cau.edu.cn (B. Wang), s20233081802@cau.edu.cn (R. Ai), ghy@cau.edu.cn (G. Yu), S20223081677@cau.edu.cn (Y. Ren), liujincun@cau.edu.cn (J. Liu), wyg@cau.edu.cn (Y. Wei), andong@cau.edu.cn (D. An), yan.meng@cau.edu.cn (Y. Meng).

2020; Yan, Wang, Wu, Tan, & Yu, 2023; Yu, Sun, Xu, & Tan, 2015), but there remains a significant gap compared to real fish. However, target tracking for robotic fish, as one of the key issues in achieving various application tasks, requires environmental perception of the target, which guides motion control to ultimately achieve dynamic target following. For example, Hu, Zhao, Wang, and Jia (2009) integrated a digital camera into the robotic fish and used a modified continuous adaptive mean shift approach to obtain visual target information under degraded image conditions. Embedded hardware was installed in the robotic fish by Sun, Yu, Chen, and Xu (2014), and a color and shape recognition algorithm, which synthesizes the CAMSHIFT algorithm and the Kalman filter, was proposed to achieve real-time visual target localization. Yu, Wu, Yang, Yang, and Zhang (2020) utilized a miniature camera to take underwater pictures and applied the kernelized correlation filter (KCF) for visual target tracking on a robotic fish, although KCF requires initial labeling and makes it difficult to re-identify the target once it is lost. Ji et al. (2020) used a visual sensor Pixy CMUcam5 to capture the position and size of a specific color target and subsequently made motion decisions based on this information. A color-based dynamic underwater target binocular positioning framework and a robust target tracking controller for underwater target tracking in robotic manta rays were proposed by Meng et al. (2023). Although the aforementioned studies have enabled visual target perception and tracking in robotic fish, there are still several limitations in vision-based perception and processing methods.

The first issue is image degradation. The onboard camera mounted on the fish head will unavoidably shake when the bionic robotic fish moves, resulting in underwater imaging motion blur and ghosting. Additionally, the movement of the target object itself can also cause image blurring. This combined blur in underwater target imaging further degrades the effectiveness of target perception algorithms. Currently, most existing research aims to mitigate motion blur through gimbal-based motion compensation or electronic image stabilization methods. For example, Yang, Wu, and Yu (2016) designed a camera stabilizer for visual systems based on an Inertial Measurement Unit and a feedback-feedforward controller, achieving more stable images. A framework of estimation and prediction for the real-time digital video stabilization of bionic robotic fish was proposed by Meng, Wu, Zhang, Wang, and Yu (2022), achieving approximate visual stability and significantly faster stabilization speed. Although these studies achieved favorable experimental results, physical and electronic stabilization methods optimize at the expense of the robotic fish's field of view and pose challenges in directly obtaining accurate target location information. Consequently, it is thought that using image processing algorithms to enhance the clarity of motion-blurred images is another practical solution. To avoid the issue of sacrificing the field of view when performing motion deblurring using physical and electronic stabilization, and considering the rapid development of edge computing technologies and deep learning-based image motion deblurring methods, real-time motion blur image quality improvement using deep learning-based image processing algorithms on the edge computing side has become an effective and feasible solution. Currently, mainstream deep learning-based motion deblurring methods achieve excellent deblurring results, with most related research focusing on designing more complex models to further improve the deblurring metrics of the algorithm. However, the improvement in motion deblurring performance often comes at the cost of high computational expenses (Zhang et al., 2022). Advanced algorithms with superior motion deblurring performance are typically deployed on high-end GPU desktops or servers, which results in poor real-time performance and makes edge computing deployment unfeasible. Moreover, there is very limited research on lightweight deep learning-based motion deblurring methods. Therefore, it is crucial to research deep learning-based motion deblurring algorithms that efficiently balance deblurring performance and computational efficiency, enabling the algorithms to run in a lightweight and real-time manner, as well as supporting deployment on edge computing platforms. A machine fish platform equipped with

a lightweight deep learning-based motion deblurring algorithm can effectively capture clearer and more reliable images for subsequent visual tasks, without sacrificing the field of view.

In the vision-based target tracking tasks of robotic fish, in addition to image degradation issues, there are also robustness problems in target perception algorithms for image processing. Traditional image processing-based methods for target color and shape recognition and tracking algorithms have the advantage of low computational cost but suffer from poor portability and robustness, making it difficult to re-identify the target after it is changed or lost. With the advancement of edge computing technology and the enhancement of device computing power, it has become feasible for robotic fish to carry edge computing devices to implement deep learning-based image processing and visual perception. For example, Han et al. (2023) achieved target position detection by equipping a robotic fish with an edge computing device and deploying the YOLOX target detection algorithm, providing the required directional information to the controller to achieve 2D tracking. Although deep learning-based image processing target detection methods require higher computational power compared to traditional methods, some one-stage target detection algorithms, such as the YOLO series (Bochkovskiy, Wang, & Liao, 2020; Redmon, Divvala, Girshick, & Farhadi, 2016; Redmon & Farhadi, 2017, 2018), can achieve real-time operation on edge computing devices with strong robustness and high detection accuracy, providing more reliable algorithmic support for target perception and localization in robotic fish.

Research on real-time motion deblurring algorithms to provide reliable data for subsequent tasks is of practical significance. It helps improve the quality of motion-blurred images in marine robots when performing tasks such as autonomous visual target tracking, particularly enhancing the accuracy and stability of systems that rely on reliable visual information for real-time decision-making. For example, it can improve the accuracy of marine robot target detection systems, providing solutions for more precise target recognition and tracking in marine environments. In the surveillance field, especially at the edge computing level, it enhances the ability to track and identify high-speed moving targets by reducing motion blur. In the environmental monitoring field, particularly in aquatic species and biodiversity monitoring, it improves the accuracy of data collection, providing more reliable image data for scientific research and decision-making.

Therefore, this paper aims to design a lightweight motion deblurring method that can run in real-time on edge computing devices, providing enhanced images for autonomous visual target tracking of biomimetic robotic fish. Additionally, it integrates the lightweight motion deblurring method with a lightweight object detection algorithm to improve detection accuracy and enable more efficient target position perception. The algorithm was deployed and experimentally validated on the robotic tuna platform. The primary contributions of this paper are as follows:

1. A lightweight deep learning image motion deblurring algorithm (LMDNet) is proposed in this paper, which achieves better results on the public GOPRO dataset (Nah, Hyun Kim, & Mu Lee, 2017) compared to other lightweight deep learning motion deblurring algorithms.
2. Using a small robotic shark as the detection target, the joint deployment of the proposed motion deblurring algorithm and the Yolov5 target detection algorithm on edge computing devices is implemented in this paper to obtain target location information.
3. Based on the acquired target position (targeted at the small robotic shark), this paper implements motion control on the robotic tuna platform to achieve autonomous visual target tracking on a two-dimensional plane, with experimental validation performed accordingly.

The main challenge in this study lies in how to achieve better motion deblurring performance with a lightweight model architecture, and how to balance computational efficiency with motion deblurring performance.

In most cases, changes to the model cannot yield performance gains and may even break the performance balance of the model, leading to performance degradation. The design and modifications of the model must focus on optimizing the feature extractor's ability, maximizing performance gains with the most cost-effective computational cost. At the same time, it is crucial to consider whether the theoretical innovations in the model can be practically deployed in real-world engineering applications. Furthermore, this study also involves significant engineering applications and deployment costs. Therefore, the design of a lightweight real-time motion deblurring algorithm and its experimental implementation on the machine fish platform will, to some extent, fill the gap in the lightweight design and deployment application direction of motion deblurring algorithms.

The remainder of this paper is structured as follows: In Section 2, the related works in the field of motion deblurring are first summarized, followed by an overview of the electromechanical structure of the robotic tuna platform used in this study, as well as the hardware design of the head mechanism. The section also highlights the image motion blur caused by factors such as the robotic fish's movements. Section 3 discusses the architecture of the lightweight motion deblurring network (LMDNet) and its loss function. Algorithm validation of the LMDNet network is conducted in Section 4, where improvements in Yolov5 target detection performance through LMDNet preprocessing are demonstrated. Section 5 presents the experimental implementations of autonomous visual target tracking utilizing the LMDNet algorithm. Section 6 discusses the experimental summary of the proposed algorithm, the scalability of the research, and the limitations of the study. Section 7 provides a summary of the entire paper.

2. Related works and the robotic tuna platform

2.1. Related works

Image motion deblurring is a classic problem in low-level computer vision, aimed at restoring a blurred image caused by motion into a sharp one. Image motion deblurring methods are divided into non-deep learning-based methods and deep learning-based methods. Non-deep learning-based image motion deblurring methods typically treat the deblurring task as an inverse filtering problem, where the blurred image is considered the result of a convolution with a blur kernel. Therefore, non-deep learning-based methods can be categorized into two scenarios: known blur kernel and unknown blur kernel. Early methods assumed the blur kernel was known and used image deconvolution algorithms to restore the image. For example, Schmidt, Rother, Nowozin, Jancsary, and Roth (2013) used synthetically generated blur kernels to create training data and proposed a non-blind deblurring method for image restoration. Xu, Tao, and Jia (2014) introduced an effective spatial deconvolution method that combines sparse priors to suppress noise and visual artifacts. Additionally, blind image motion deblurring assumes the blur kernel is unknown, with the goal of simultaneously recovering both the blurred image and the blur kernel itself (Zhang et al., 2022). For example, Xu and Jia (2010) proposed an efficient kernel estimation method using spatial priors and Iterative Support Detection (ISD) kernel refinement to robustly suppress noise. Bahat, Efrat, and Irani (2017) proposed a blind image deblurring method for non-uniform blur based on an internal image-specific block recursive prior. While non-deep learning-based image motion deblurring methods perform well in some cases, they tend to perform poorly in more complex and common scenarios such as strong motion blur.

Thanks to the development of deep learning technologies, deep learning models, which have the ability to efficiently extract features from vast amounts of data, have seen rapid and widespread adoption. This has greatly facilitated advancements in the field of image motion deblurring. Deep learning-based image motion deblurring algorithms can efficiently improve the quality of motion-blurred images. For example, Mao, Li, and Wang (2024a) developed a reversible decoder named

AdaRevD to address the issue of insufficient deblurring decoding capability and obtained excellent results on the GOPRO dataset. An efficient self-attention solver based on frequency domain and a simple yet effective discriminative feed-forward network based on frequency domain (FFTformer) were proposed by Kong, Dong, Ge, Li, and Pan (2023), achieving superior deblurring results. Mao, Wang, Xie, Li, and Wang (2024b) proposed a Local Frequency Transformation (LoFormer) method, which significantly improves the performance of image deblurring tasks. Fang et al. (2023) proposed a method called UFPDeblur for normalizing the representation of motion blur kernels in the latent space, which achieves high accuracy and robustness. Ghasedabadi et al. (2024) introduced an encoder-decoder architecture called CGNet for image restoration that utilizes a global context extractor. Cui, Ren, Cao, and Knoll (2024) proposed a multi-scale wavelet transform-based image restoration algorithm named FSNet that highlights useful frequency subbands through channel-level attention weights. Gao et al. (2023) introduced a mountain-shaped design based on a basic U-Net framework and employed multi-head attention intermediate blocks to capture more global information, ultimately achieving advanced deblurring results. Kim, Kwon, and Ko (2024) proposed a novel approach, called Illuminance-robust Multi-stage DeblurNet with Channel Attention (IMDeCA), to restore the detailed structure of the face by using the semantic mask and landmark information of the face. Sharif, Naqvi, Ali, and Biswas (2023) introduced DarkDeblur, a network leveraging dense attention blocks, a contextual gating mechanism, and a multi-term objective function for effective single-shot low-light image deblurring. Considering the importance of frequency domain features, Zhang, Sun, Sun, Wang, and Zhu (2024) proposed an image deblurring method based on self-attention and residual wavelet transform.

The aforementioned deep learning-based image motion deblurring algorithms have achieved excellent deblurring performance, but they come with high computational costs. Furthermore, there are only a few lightweight deep learning image motion deblurring algorithms suitable for edge computing, and these often face issues such as lower performance or a low parameter count but high computational demands. For example, Kupyn, Martyniuk, Wu, and Wang (2019) designed a lightweight motion deblurring network, DeblurgAN-V2, which, although lightweight and capable of real-time operation, still needs improvement in its motion deblurring performance. Guo, Wang, Dai, Wang, and Li (2022) developed a lightweight motion deblurring network, LNNet, with nested connections. Although it has a small number of parameters, it still suffers from the drawback of high computational cost. Therefore, in light of the computational constraints of edge computing devices and the performance of lightweight motion deblurring algorithms, it is essential for efficient motion deblurring algorithms to be developed and deployed to further enhance real-time performance and applicability in edge scenarios. The comparison of LMDNet with existing motion deblurring algorithms is shown in Table 1.

2.2. Overview of the robotic tuna platform

The bionic robotic fish used in this study features a streamlined tuna-inspired design, which helps reduce fluid resistance during motion, with an overall length of approximately 1.15 m (Ji, Wei, Liu, & An, 2023). The conical head and fusiform body provide ample interior room to accommodate numerous components, making it convenient for the installation of different devices. The shell of the robotic fish is constructed from polyoxymethylene engineering plastic, and the prototype of the robotic tuna platform along with the head mechanism design is shown in Fig. 1.

Structurally, the bionic robotic fish is composed of four primary components: the head, body, waist, and tail. To fulfill the requirements of the visual target tracking tasks for the robotic tuna platform, a detachable head mechanism was designed, as illustrated in Fig. 1. The conical head features a transparent chamber constructed of 3-millimeter-thick acrylic. Internally, a binocular camera, a servo, and a camera bracket are installed. The binocular camera and servo are fixed to the conical

Table 1

Comparison of LMDNet with existing deblurring algorithms.

Category	Example Algorithms	Deblurring Effect	Characteristics
Non-deep learning motion deblurring approaches	Bahat et al. (2017); Schmidt et al. (2013); Xu and Jia (2010); Xu et al. (2014)	Fair	Perform poorly in complex but common scenarios such as severe motion blur.
Deep learning motion deblurring approaches	AdaRevD (Mao et al., 2024a); FFTformer (Kong et al., 2023); LoFormer (Mao et al., 2024b); UFPDeblur (Fang et al., 2023); M3SNet (Gao et al., 2023) et al.	Excellent	Utilize complex network structures, leading to high computational cost, particularly when based on Transformer architecture.
	DeblurGAN-V2 (Kupyn et al., 2019)	Fair	Fewer model parameters and lower computational cost, but with fair performance.
	LNNNet (Guo et al., 2022)	Good	Fewer model parameters, but high computational cost.
	LMDNet (Ours)	Good	Fewer model parameters and lower computational cost, with the ability to be deployed on edge computing devices.

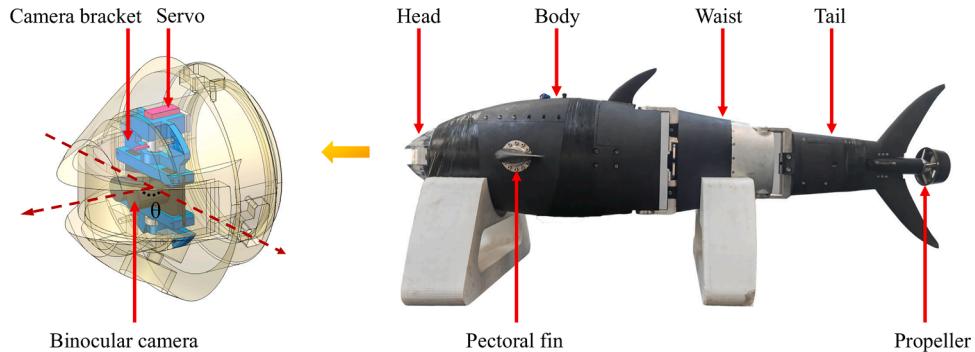


Fig. 1. Prototype of the robotic tuna platform and head mechanism design diagram.

head through the camera bracket, and the servo is connected to the binocular camera through the same bracket, enabling the camera to rotate along with the servo's movement. The servo has a rotation angle of 180°, making the θ angle range from 0 to 180° as shown in Fig. 1. The camera itself has a viewing angle of 90°. When the binocular camera rotates via the servo, the maximum field of view (FOV) of the robotic tuna reaches 270°. Thus, the wider FOV provided by the head mechanism is beneficial for target searching. Additionally, the body section of the robotic tuna is equipped with circuit boards, pectoral fin servos, lithium battery packs, and a motor for the waist joint. The waist houses the motor for the tail joint along with its corresponding drive circuit board. The tail is fitted with a propeller and a propeller drive circuit board. Therefore, the robotic tuna's motion control module consists of the pectoral fins, waist-tail joints, and tail propeller, enabling it to perform forward swimming, backward swimming, and turning movements through its control system. Table 2 shows the technical specifications for the robotic tuna.

The swimming and turning of the robotic tuna inevitably cause head shaking. The image sequences showing head motion induced by the movement of the robotic tuna are presented in Fig. 2, with a total du-

ration of 1 second. Additionally, when the head camera searches for a target, it rotates using the servo. Moreover, the tracked target itself is also moving. These factors collectively contribute to the motion blur in the images acquired by the camera. Fig. 3 shows the underwater clear and motion-blurred pictures acquired by the robotic tuna's camera. Column a shows the clear images captured when the robotic fish is stationary, while columns b and c display blurred images captured during its motion. The loss of target features due to artifacts in blurry images makes it challenging to fix the problem by improving target detection algorithms solely. Therefore, this study focuses on motion deblurring algorithms to improve image quality, thereby increasing the success rate of subsequent target detection.

3. Lightweight motion deblurring network - LMDNet

Currently, image processing algorithms based on Transformer (Vaswani et al., 2017) and Mamba (Gu & Dao, 2023) architectures are gaining significant research interest. However, they have high computational costs and are challenging to deploy on edge computing devices. In contrast, convolutional neural network (CNN) architectures have an advantage in edge computing deployments. Therefore, a lightweight motion deblurring network, LMDNet, based on a CNN structure, is proposed. The basic network structure is constructed using the U-Net (Ronneberger, Fischer, & Brox, 2015) architecture and NAF-Block (Chen, Chu, Zhang, & Sun, 2022c) modules. Secondly, to ensure consistency between feature extraction during inference and training for enhanced inference performance, the Test-time Local Converter (TLC) (Chu, Chen, Chen, & Lu, 2022) is employed for global feature aggregation. In addition, the Ghost module (Han et al., 2020) is incorporated into the encoder of the base network to create more feature maps at a reduced computational cost, thereby improving feature extraction. Finally, a wavelet loss function tailored for image motion deblurring is proposed, further enhancing the network's deblurring performance.

Table 2

Technical specifications of the robotic tuna.

Technical specifications	Characteristics
Dimensions (L × W × H)	1.15 × 0.562 × 0.393m ³
Total weight	19 kg
Number of motors	4
Drive type	DC brushless motor
Power supply	DC 24 V
Main controller	STM32F407
Edge computing device	Nvidia Jetson Xavier NX (8G)
Visual range	270°/5 m

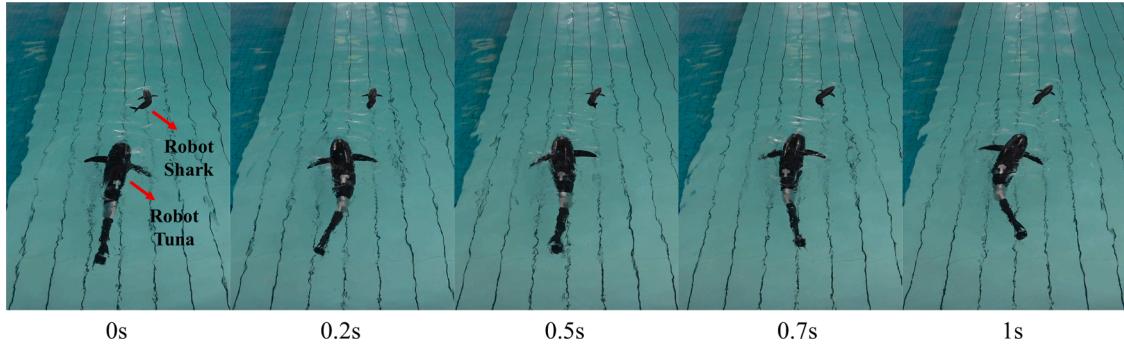


Fig. 2. Image sequence showing head motion induced by the movement of the robotic tuna.

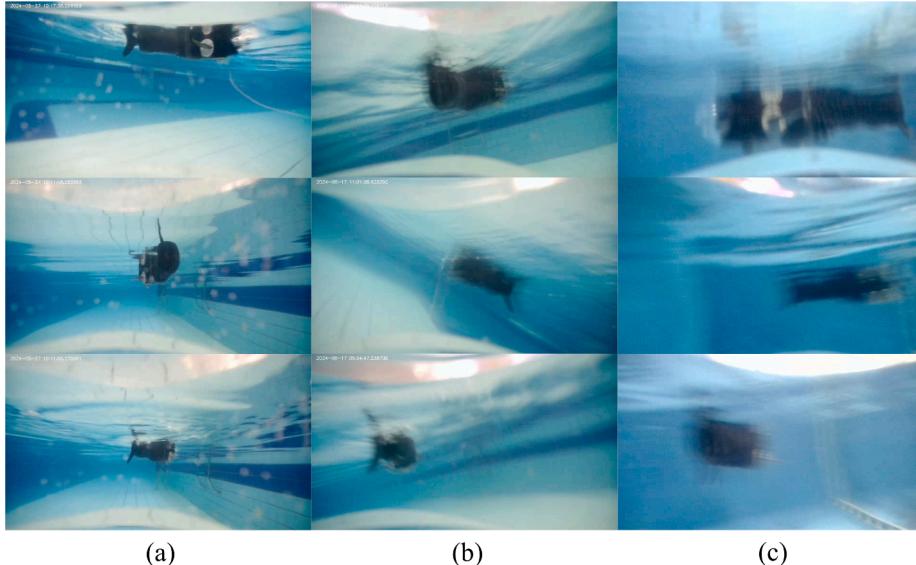


Fig. 3. Underwater clear images and motion-blurred images captured by the robotic tuna's camera. Column a shows clear images taken when the robotic tuna is stationary, while columns b and c display the blurred images captured during the motion of the robotic tuna.

3.1. Overall network architecture

The overall architecture of the lightweight motion deblurring network LMDNet is shown in Fig. 4. The main network architecture, marked by the red dashed box in Fig. 4, primarily adopts the U-Net encoder-decoder structure and NAF-Block modules. Unlike many SOTA motion deblurring networks (Kong et al., 2023; Wang et al., 2022a) with high system complexity and computational cost, the NAFNet network simplifies the baseline using a straightforward convolutional neural network structure. This algorithm achieves notable motion deblurring performance, and the NAF-Block in its structure is highly conducive to lightweight optimization and deployment on edge computing devices. Therefore, this paper adopts NAF-Block as the fundamental component for the network's encoder and decoder. In the four encoding layers of the encoder, the number of NAF-Blocks used is 1, 1, 1, and 14, respectively, while the middle layer contains 1 NAF-Block. In the four decoding layers of the decoder, each layer also utilizes 1 NAF-Block. The width of the network is set to 16, which ensures lightweight operation while maintaining baseline motion deblurring performance, defined in this paper as the Baseline. In other words, the Baseline refers to a network in which the number of convolution output channels in the NAFNet32 (Chen et al., 2022c) network is reduced from 32 to 16, and the number of NAF-Blocks in the fourth encoding layer of the NAFNet32 network is reduced from 28 to 14. Additionally, the Baseline can be understood as a simplified framework shown in the overall architecture diagram of LMDNet in Fig. 4, where all Ghost modules are removed, the wavelet

loss function used during training is omitted, and the TLC method used during inference is also excluded.

The overall structure of the NAF-Block is shown in Fig. 4, which includes two special components: the Simple Channel Attention (SCA) structure and the SimpleGate structure. Because nonlinear activation functions increase computational complexity, both SCA and SimpleGate are designed to avoid the use of nonlinear activations. The SCA module replaces the multilayer perceptrons in the Channel Attention (CA) (Hu, Shen, & Sun, 2018) module, after spatial information compression (pooling operation), with a single convolutional layer. The structures of CA and SCA are shown in Fig. 5. Compared to CA, SCA reduces the computational cost of nonlinear activations without sacrificing algorithm performance (Chen et al., 2022c). SimpleGate performs feature map splitting and element-wise multiplication operations, acting as a substitute for the activation function in the module and enhancing algorithm performance (Chen et al., 2022c). The construction of SimpleGate is illustrated in the lower right corner of Fig. 4.

3.2. Test-time local converter

The training of motion deblurring models is often constrained by GPU memory limitations. As a result, input images during the training phase are cropped from high-resolution images into smaller-resolution images, while the trained model is directly used to restore high-resolution images during inference. Global operations in the model, such as global average pooling, can cause inconsistencies in feature

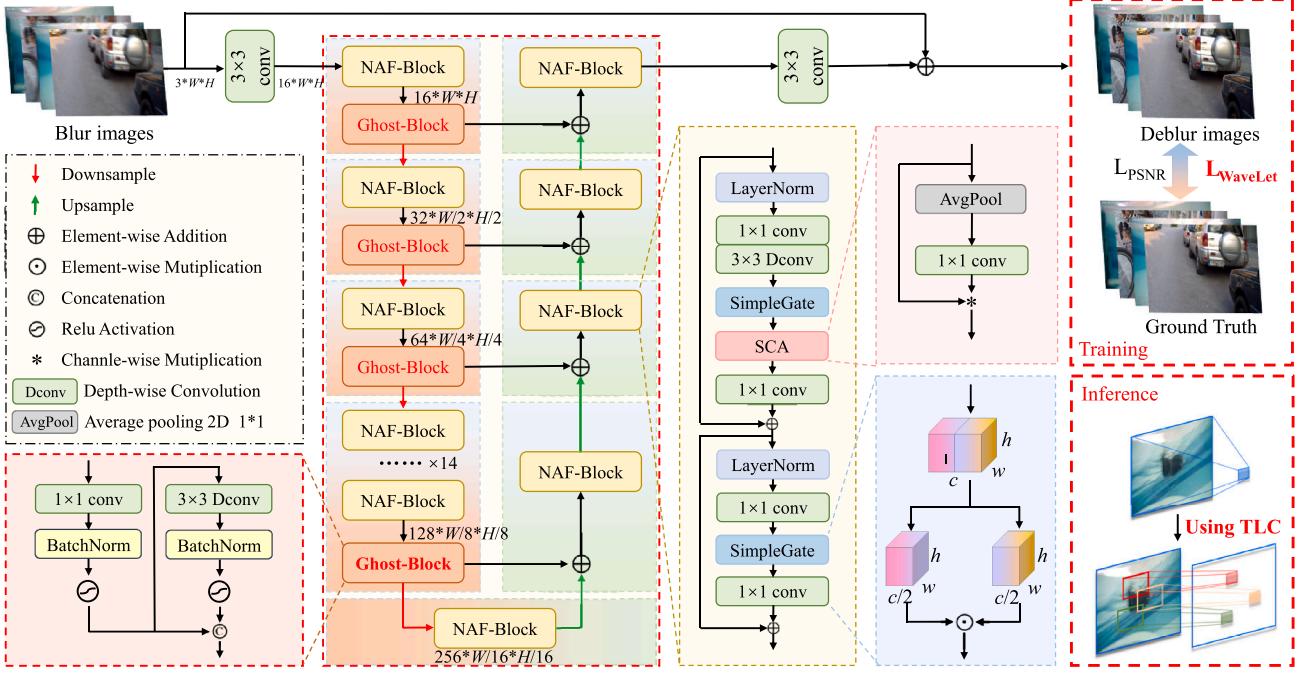


Fig. 4. Overall architecture of the lightweight motion deblurring network LMDNet.

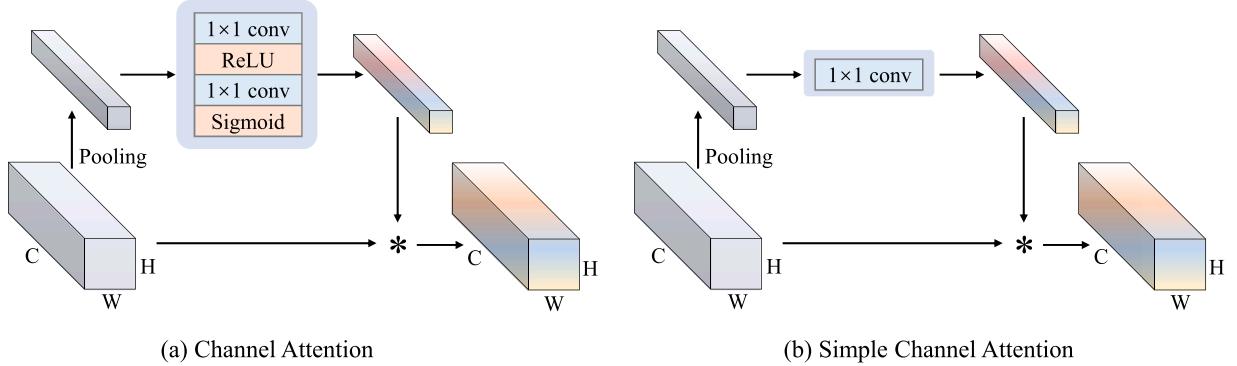


Fig. 5. Structural diagram of the CA module and the SCA module.

aggregation due to differences in image sizes used during training and inference (local image regions during training, but the entire image during inference). This inconsistency in block-based feature distribution negatively impacts model performance. To address this issue of inconsistent feature distributions and the resulting drop in deblurring performance during the test phase, this paper incorporates TLC (Chu et al., 2022) in the network. The training and inference processes of the motion deblurring network using TLC are illustrated in Fig. 6. The core idea of TLC is to replace global information aggregation, such as global average pooling, with a local information aggregation approach at the inference stage. Specifically, for a feature map $X \in \mathbb{R}^{H \times W}$, the aggregation operation $\Phi(X, f)$ is defined in Eq. 1.

$$\Phi(X, f) = \frac{1}{H \times W} \sum_{p=1}^H \sum_{q=1}^W f(X_{p,q}) \quad (1)$$

where f is the aggregation function, such as mean or sum. The corresponding local aggregation $\Psi(X, f)$ for each pixel (i, j) in a local window of size $K_h \times K_w$ is defined in Eq. 2.

$$\Psi(X, f)_{i,j} = \frac{1}{K_h K_w} \sum_{p=1}^{K_h} \sum_{q=1}^{K_w} f(X_{p,q}) \quad (2)$$

where (p, q) are the indices within the local window centered around pixel (i, j) . This local aggregation ensures that the statistics used during inference are consistent with the patch-based aggregation during training, thus reducing the distribution shift that typically harms performance. TLC can improve restoration quality without the need for re-training or fine-tuning. The training process uses the GOPRO dataset, while the inference process uses motion-blurred robotic shark images. The TLC method converts global operations into local operations during inference by segmenting input features into overlapping windows of the same size, and independently performing information aggregation on each overlapping window to achieve global feature aggregation. The optimization using TLC results in only a 1% to 5% increase in processing time, but it significantly enhances performance.

3.3. The ghost module

To further enhance the network's feature extraction capability and generate more feature maps at a lower computational cost using efficient operations, the Ghost-Block module (Han et al., 2020) is incorporated into the four encoding layers of the network's encoder. The structure of the Ghost module is shown in Fig. 7. In each encoding layer, the Ghost-Block module reduces the redundancy of feature maps output by

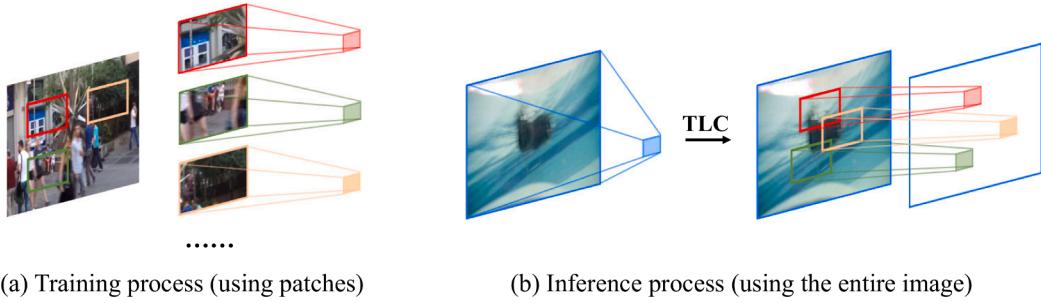


Fig. 6. Diagram of the training process and TLC-based inference process of the motion deblurring network.

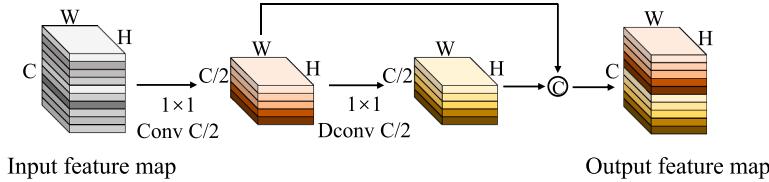


Fig. 7. Structure diagram of the Ghost module. The stride of the Ghost-Block module is set to 2.

the NAF-Block module while generating ghost feature maps that effectively capture the intrinsic feature information. Mathematically, for the input feature map $X \in \mathbb{R}^{C \times H \times W}$, where C is the number of input channels, and H and W are the height and width respectively, the convolution operation can be represented by Eq. 3.

$$Y = X * f + b \quad (3)$$

where $*$ denotes the convolution operation, $f \in \mathbb{R}^{C \times k \times k \times N}$ are the convolution filters, and N is the number of output feature maps. The number of floating point operations (FLOPs) for this operation is $N \times H' \times W' \times C \times k \times k$, where H' and W' are the height and width of the output feature map. To reduce computational cost, the Ghost module adopts a two-step operation: the primary convolution and the linear operation. The primary convolution operation generates a small number of intrinsic feature maps using fewer filters, denoted as $Y' \in \mathbb{R}^{M \times H' \times W'}$. The computational formula for this is shown in Eq. 4.

$$Y' = X * f' \quad (4)$$

where $f' \in \mathbb{R}^{C \times k \times k \times M}$, $M \leq N$, and M is the number of intrinsic feature maps. The linear operation expands these intrinsic feature maps to generate the final output feature maps. Each intrinsic feature map y'_i is processed by a linear operation δ to generate S ghost features, as given by Eq. 5.

$$y_{ij} = \delta_{i,j}(y'_i), \quad \forall i = 1, \dots, M, j = 1, \dots, S \quad (5)$$

where $\delta_{i,j}$ represents the j -th linear operation applied to the i -th intrinsic feature map y'_i . The resulting set of ghost feature maps, $Y = [y_{11}, y_{12}, \dots, y_{MS}]$, provides $N = M \times S$ output feature maps. The Ghost module generates more feature maps with low-cost linear operations, enabling it to more fully exploit the high-dimensional information in the input feature maps. In the proposed network, the stride of the Ghost-Block module is set to $S = 2$.

3.4. The wavelet loss

Considering the characteristics of motion deblurring tasks, where the blurred boundaries in images often correspond to distinguishable frequency signals, this paper proposes a loss function based on discrete wavelet transform, called $Loss_{Wavelet}$. Backpropagation and parameter updates are performed by comparing the frequency domain differences between the processed images and the ground-truth images during training. Specifically, the wavelet loss is defined as the sum of the absolute

differences between the wavelet-transformed images of the inferred outputs and the ground-truth images. In this study, the Haar wavelet is used for wavelet transformation. The Haar scaling function and Haar wavelet function are shown in Eqs. 6 and 7, respectively.

$$\varphi(a) = \begin{cases} 1 & \text{if } 0 \leq a < 1 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$\psi(a) = \begin{cases} 1 & 0 \leq a < 0.5 \\ -1 & 0.5 \leq a < 1 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

The 2D scaling function and 2D wavelet functions are shown in Eq. 8. Here, $\varphi(a)$ represents the 1D scaling function, $\psi(a)$ is the 1D wavelet function, $\varphi(a, b)$ is the 2D scaling function, and $\psi^H(a, b)$, $\psi^V(a, b)$, and $\psi^D(a, b)$ are 2D wavelet functions corresponding to variations along the horizontal, vertical, and diagonal edges, respectively.

$$\begin{aligned} \varphi(a, b) &= \varphi(a)\varphi(b) \\ \psi^H(a, b) &= \psi(a)\varphi(b) \\ \psi^V(a, b) &= \varphi(a)\psi(b) \\ \psi^D(a, b) &= \psi(a)\psi(b) \end{aligned} \quad (8)$$

The scaling and translation basis functions are defined in Eqs. 9 and 10. The discrete wavelet transform formulas for an image of size $X \times Y$ are shown in Eqs. 11 and 12, where $W_\varphi(0, x, y)$ represents the low-frequency sub-band of image $f(a, b)$, and $W_\psi^H(j, x, y)$, $W_\psi^V(j, x, y)$, and $W_\psi^D(j, x, y)$ denote the high-frequency sub-bands in three different directions. In this study, the wavelet decomposition level is set to $J=4$, so the value of j ranges over 0, 1, 2, 3.

$$W_\varphi(0, x, y) = 2^{j/2}\varphi(2^j a - x, 2^j b - y) \quad (9)$$

$$W_\psi^i(j, x, y) = 2^{j/2}\psi^i(2^j a - x, 2^j b - y), \quad i = \{H, V, D\} \quad (10)$$

$$W_\varphi(0, x, y) = \frac{1}{\sqrt{XY}} \sum_{a=0}^{X-1} \sum_{b=0}^{Y-1} f(a, b) \varphi_{0,x,y}(a, b) \quad (11)$$

$$W_\psi^i(j, x, y) = \frac{1}{\sqrt{XY}} \sum_{a=0}^{X-1} \sum_{b=0}^{Y-1} f(a, b) \psi_{j,x,y}^i(a, b), \quad i = \{H, V, D\} \quad (12)$$

The wavelet loss function defined in this paper is shown in Eq. 13, where $W_{\varphi-DB}$ and $W_{\psi-DB}^i$ represent the low-frequency and high-frequency sub-bands of the motion deblurred image, respectively, while

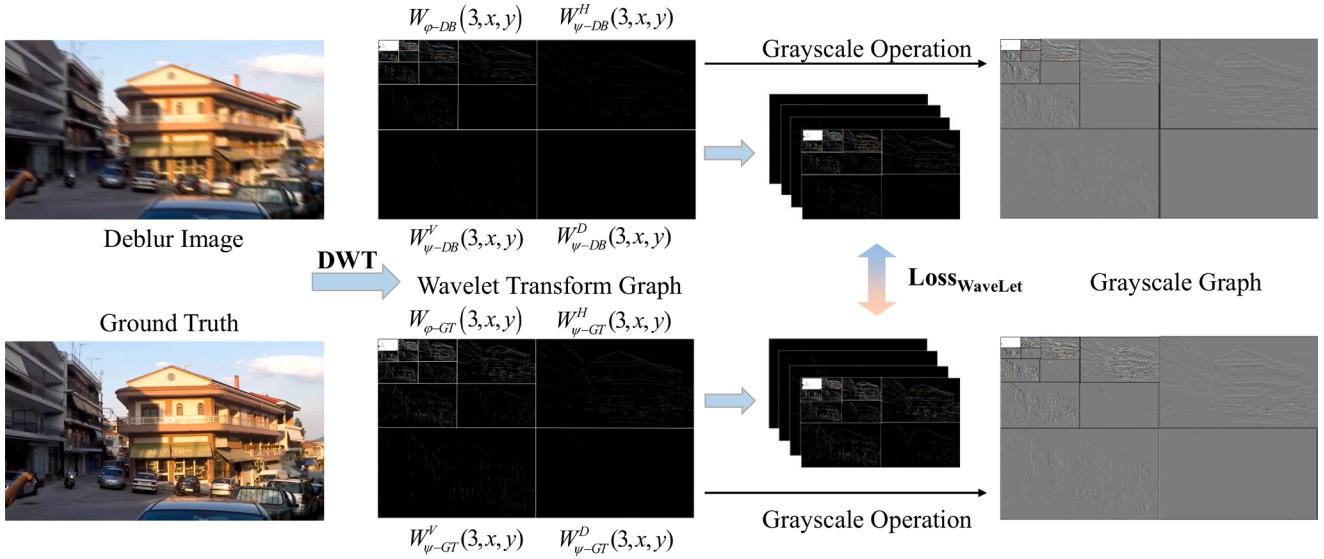


Fig. 8. Workflow diagram of wavelet loss calculation.

$W_{\varphi-GT}$ and $W_{\psi-GT}^i$ denote the low-frequency and high-frequency subbands of the ground-truth image, respectively. This paper also employs PSNR loss, and the formula for the model's loss function is presented in Eq. 14. Considering that multiple loss functions involve weight allocation, weight a is set to 1, while weight b is set to 1000.

$$\begin{aligned} \text{Loss}_{\text{WaveLet}} &= |W_{\varphi-DB}(0, x, y) - W_{\varphi-GT}(0, x, y)| \\ &+ \sum_{i=H,V,D} \sum_{j=0}^{J-1} |W_{\psi-DB}^i(j, x, y) - W_{\psi-GT}^i(j, x, y)| \quad (13) \end{aligned}$$

$$\text{Loss} = a * \text{Loss}_{\text{PSNR}} + b * \text{Loss}_{\text{WaveLet}} \quad (14)$$

To further reduce the impact of the increased computational cost from image wavelet transformations on training time, the process is implemented using Pytorch-based wavelet transform (Cotter, 2020) in this study. The workflow of the wavelet loss calculation in this study is illustrated in Fig. 8. To emphasize the high-frequency information in the wavelet-transformed images, grayscale processing was applied to generate the grayscale images shown in the figure.

4. LMDNet algorithm validation experiment

4.1. Motion deblurring experiment dataset and training details

The motion deblurring dataset used in this paper is the GOPRO dataset (Nah et al., 2017), which consists of 3214 pairs of motion-blurred images and corresponding clear images with a resolution of 1280×720 . Among them, 2103 pairs were used for training, and 1111 pairs were used for testing. The 1111 pairs of images in the GOPRO dataset's test set cover various blurred and deblurred image pairs from different scenes in the training dataset. Therefore, the data distribution of the GOPRO test dataset is consistent with that of the GOPRO training dataset. The patch size of the training dataset is empirically set to 256×256 pixels (Chen et al., 2022c; Fang et al., 2023; Kong et al., 2023), and the batch size is set to 24. The patching strategy is the same as in NAFNet (Chen et al., 2022c). Specifically, the training dataset is divided into 512×512 pixel images with a step size of 256. During training, patches of size 256×256 pixels are randomly cropped from the 512×512 pixel images, along with random flipping and rotation augmentations. The inconsistency in feature aggregation when inferring the entire image in the test dataset caused by patching the training dataset is addressed in this paper by using the TLC (Chu et al., 2022) method. During model training, AdamW was used as the optimizer, with an initial

learning rate of 10^{-3} and a weight decay of 10^{-3} . No pre-trained weights were used during model training. Instead, network weight parameters were randomly initialized using a random seed, and the model was re-trained for a total of 1500 epochs. The experimental hardware includes an Intel(R) Xeon(R) E5-2650v4 @ 2.20GHz \times 12 CPU and four NVIDIA GTX 2080ti GPUs.

4.2. Performance comparison of LMDNet for motion deblurring

In this research, the network that solely employs the NAF-Block is referred to as the Baseline. The algorithm performance of Baseline, after incorporating TLC, wavelet loss function, and Ghost-Block module, was validated, and the comparison results of the LMDNet motion deblurring algorithm on the GOPRO dataset are presented in Table 3. The PSNR and SSIM metrics in this paper are the average values of image pair differences in the GOPRO test dataset, with an image resolution of 1280×720 pixels. As shown in the table, the Baseline model achieved a Peak Signal-to-Noise Ratio (PSNR) of 30.958 dB and a Structural Similarity Index Measure (SSIM) of 0.9427 on the GOPRO dataset, with only 2.68M parameters and a computational cost of 12.66 Giga Multiply-Accumulate Operations (GMACs). Using Baseline as the foundational network for improvements, the model achieved adequate motion deblurring performance while maintaining low computational cost and minimal parameters, making it suitable for lightweight deployment in future experiments. By incorporating the LossWavelet method, the PSNR of the Baseline + LossWavelet network improved by 0.381 dB, while the model's parameter size and computational cost did not increase, indicating that the Haar wavelet loss function can enhance the motion deblurring performance of the model without adding extra computational overhead. By incorporating the TLC method, the Baseline + TLC network increased PSNR by 0.4647 dB, with the computational cost rising from 12.66 GMACs to 14.37 GMACs (an increase of 1.71 GMACs), while the number of parameters remained unchanged. Compared to the Baseline + TLC network, the Baseline + TLC + Ghost network, which incorporated the Ghost-Block module in the encoder, further enhanced model performance. The PSNR was improved by 0.1702 dB, while the number of parameters and computational cost only increased by 0.02M and 0.23 GMACs, respectively. The LMDNet network, which was defined as Baseline + TLC + Ghost + LossWavelet, achieved a PSNR improvement of 0.7444 dB (an increase of 2.4 %) compared to the Baseline by combining TLC and Ghost-Block and introducing the wavelet loss function during training. The number of parameters and computational cost only increased by 0.02M and 1.94GMACs.

Table 3

Comparison of motion deblurring algorithm results for LMDNet on the GOPRO dataset. Computational complexity is calculated based on images with a resolution of 640×480 .

Algorithms	TLC	Ghost	LossWaveLet	PSNR	SSIM	Parameter(M)	MACs(G)
Baseline				30.9580	0.9427	2.68	12.66
Baseline + LossWavelet			✓	31.3390	0.6456	2.68	12.66
Baseline + TLC	✓			31.4227	0.9471	2.68	14.37
Baseline + TLC + Ghost	✓	✓		31.5929	0.9483	2.7	14.6
LMDNet	✓	✓	✓	31.7024	0.9498	2.7	14.6



Fig. 9. Comparison of different motion deblurring algorithms on the GOPRO dataset.

Fig. 9 shows the quality evaluation of the LMDNet motion deblurring algorithm on the GOPRO dataset. The first column of the image contains the original blurry images, while the second, third, and fourth columns display the images processed by the Baseline, Baseline + LossWavelet, and LMDNet models, respectively. The two sub-images below each image show the enlarged details of the areas marked by the red and green boxes in the image. The PSNR values under each image represent the PSNR calculation between the processed image and the ground truth clear image. From the three sets of images in Fig. 9, it is evident that Baseline + LossWavelet outperforms Baseline in terms of motion deblurring performance. The PSNR values of the deblurred images processed by the Baseline + LossWavelet model are higher than those processed by the Baseline model. Additionally, the details in the images are clearer, and ghosting noise is reduced. Compared to both Baseline and Baseline + LossWavelet, LMDNet achieves the highest PSNR values and shows the clearest image quality in all three sets of images.

4.3. Performance comparison with several motion deblurring algorithms

The LMDNet network was compared with various motion deblurring algorithms including state-of-the-art algorithms on the GOPRO dataset, and the comparison results were presented in Table 4. As Table 4 illustrates, compared to the lightweight motion deblurring algorithm DeblurGAN-V2 (MobileNetDSC), LMDNet had a higher number of parameters and computational cost but achieved better PSNR and SSIM results. LMDNet used less parameters and computing power than DeblurGAN-V2 (MobileNet) to produce greater PSNR (an increase of 3.53 dB) and SSIM (an increase of 0.025). Compared to LNNet, LMDNet achieved nearly the same PSNR and SSIM values while having only about 1/2 the number of parameters and 1/23 the computational cost of LNNet. LMDNet's parameters and computational cost were more favorable than those of NAFNet-32 and M3SNet-32, although it had somewhat lower PSNR and SSIM values. As a network of Transformer architecture, FFTformer achieved the good results in PSNR and SSIM values

Table 4

Comparison of different motion deblurring algorithms on the GOPRO dataset. Computational complexity is calculated based on images with a resolution of 640×480 .

Algorithms	PSNR	SSIM	Parameter(M)	MACs(G)
DeblurGAN-V2 (MobileNetDSC) (Kupyn et al., 2019)	28.03	0.922	2.03	4.68
DeblurGAN-V2 (MobileNet) (Kupyn et al., 2019)	28.17	0.925	3.13	20.57
LNNNet (Guo et al., 2022)	31.70	0.951	5.94	336.65
LMDNet	31.70	0.950	2.70	14.60
NAFNet-32 (Chen et al., 2022c)	32.87	0.961	17.11	86.76
M3SNet-32 (Gao et al., 2023)	32.91	0.965	16.73	88.36
MPRNet (Zamir et al., 2021)	32.66	0.959	20.13	3645.38
FSNet (Cui et al., 2024)	33.29	0.963	13.28	521.54
LoFormer-S (Mao et al., 2024b)	33.73	0.966	16.38	221.01
CGNet (Ghasemabadi et al., 2024)	33.77	0.968	119.22	291.87
UFPDeblur (Fang et al., 2023)	34.06	0.968	5.85	553.16
LoFormer-L (Mao et al., 2024b)	34.09	0.969	49.03	594.01
FFTformer (Kong et al., 2023)	34.21	0.969	16.56	616.17
AdaRevD (Mao et al., 2024a)	34.60	0.972	68.40	1622.81

in the table, but the amount of computation was too large, being 42 times that of LMDNet. Additionally, as shown in **Table 4**, current state-of-the-art motion deblurring algorithms, such as UFPDeblur, LoFormer, FFTformer, and AdaRevD, although demonstrating excellent motion deblurring performance, all have extremely high computational costs, even when their parameter sizes are relatively small. Compared to the current best motion deblurring algorithm, AdaRevD, LMDNet, although slightly inferior in terms of motion deblurring performance, has a computational cost that is only 1/111th of AdaRevD. These objective conditions determine that current advanced algorithms cannot be deployed for real-time processing on edge computing devices, and algorithms with excessive computational cost can only run inference on high-performance servers.

Meanwhile, **Fig. 10** presents a comparison of several motion deblurring algorithms' outcomes using the GOPRO dataset. In the figure, the larger the area of the circular region representing an algorithm, the higher its computational cost. The closer the algorithm is positioned to the top-left corner, the smaller its parameter size and the better its deblurring performance. The LMDNet network is located in the top-left corner of **Fig. 10**. Although LMDNet has lower PSNR values compared to NAFNet-32, M3SNet-32, and FFTformer, it is located further to the left with a smaller circular area, indicating lower parameter count and computational cost. Compared to the DeblurGAN-V2 algorithm in the bottom-left corner, LMDNet exhibits better motion deblurring performance. Additionally, as shown in **Fig. 10**, although the state-of-the-art motion deblurring algorithms such as UFPDeblur, FSNet, FFTformer, LoFormer, and AdaRevD have higher PSNR values and relatively uniform

parameter distributions, the area of the circular regions they represent is significantly larger compared to LMDNet. This also highlights the advantage of LMDNet in terms of computational cost, in comparison to the more advanced algorithms.

In summary, although complex algorithms such as FFTformer exhibit excellent motion deblurring performance, their high computational demands restrict them to running on resource-rich servers, making them unsuitable for deployment on edge computing devices. LMDNet improves inference speed and reduces computational cost by sacrificing some restoration quality. Although its PSNR and SSIM scores are slightly lower than those of state-of-the-art models, this paper emphasizes the significant advantages of LMDNet in computational efficiency. LMDNet provides competitive performance at a lower computational cost, making it especially suitable for resource-constrained environments, such as edge computing devices or real-time processing tasks, and holds great potential for application. Designed with lightweight optimization in mind, LMDNet effectively balances motion deblurring performance and computational cost, providing the necessary conditions for deployment on edge computing devices.

4.4. Target detection experiment for bionic robotic tuna based on the LMDNet network

The tracked target in the robotic tuna tracking experiment is a small robotic shark, as shown in **Figs. 2** and **3**. Therefore, a corresponding target detection dataset was created for detecting the robotic shark. The dataset contains 2650 non-blurred images of the robotic fish and corresponding annotation files. A training set of 2385 images and a validation set of 265 images were generated by dividing these images and annotations into two groups at a ratio of 0.9:0.1.

The ultimate goal of this study is to achieve joint deployment and operation of the motion deblurring algorithm and target detection algorithm on the Jetson Xavier NX edge computing device, and to perform target tracking control of the robotic tuna through visual target information. Therefore, a one-stage target detection algorithm was selected for detecting the robotic shark. The algorithms YOLOv5m, YOLOv8s, and YOLOv8m were selected and tested on the robotic shark dataset, with 200 training epochs. **Table 5** displays the outcomes of the experiment. YOLOv5m achieved the highest AP50 value on the robotic shark dataset, and its parameter count and computational cost are both lower than those of YOLOv8m. **Fig. 11** depicts the YOLO algorithms' training process using the robotic shark dataset. As shown in **Fig. 11**, YOLOv5m not only converges the fastest and is the most stable algorithm during training, but it also achieves the highest AP50 value. Additionally, considering the mature deployment support for the YOLOv5 algorithm, YOLOv5m was comprehensively chosen as the target detection algorithm for this study.

In order to resolve the motion blur in images brought on by the robotic tuna's movement, this paper first applies the LMDNet for deblurring before performing target detection on the captured images. The blurred test dataset images used for inference with LMDNet prior to target detection do not have corresponding clear ground truth images, meaning they are not paired. They can only be used for LMDNet inference to achieve a certain degree of motion deblurring. This test dataset is generated by performing network inference with high-dimensional feature information learned by LMDNet on the GOPRO training dataset, completing the removal of motion blur and artifacts to a certain extent.

Table 5

Comparison of YOLO algorithm results on the robotic shark dataset. Input image size is 640×480 .

Algorithms	AP50	Parameter(M)	MACs(G)
YOLOv5m	0.994	19.89	17.95
YOLOv8s	0.985	10.65	10.8
YOLOv8m	0.992	24.70	29.50

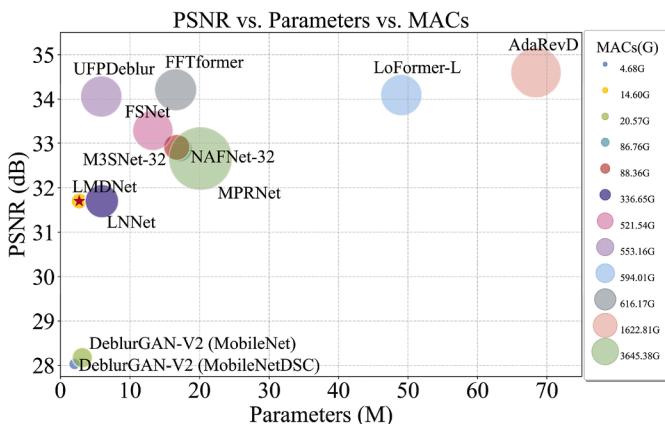


Fig. 10. Comparison of different motion deblurring algorithms on the GOPRO dataset.

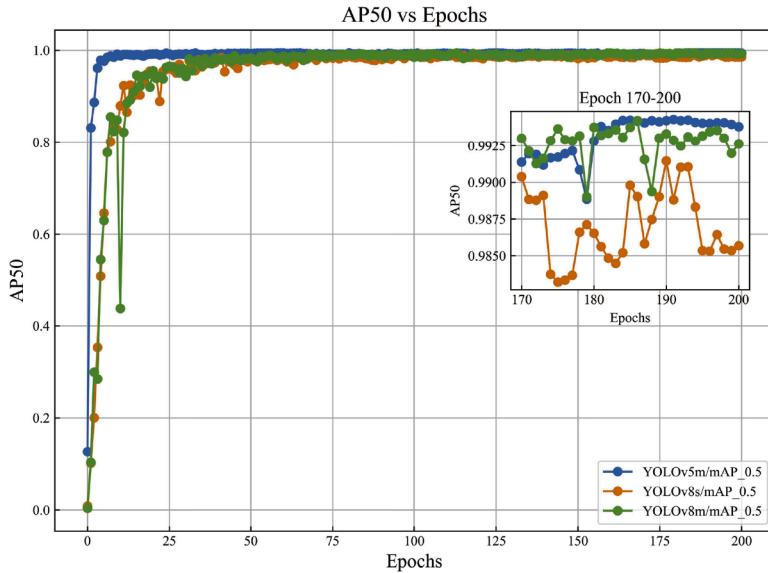


Fig. 11. Training process of YOLO algorithms on the robotic shark dataset.

To validate the effectiveness of LMDNet in improving the accuracy of target detection in motion-blurred images in aquatic environments, this study conducted comparative experiments between LMDNet and state-of-the-art motion deblurring algorithms under different lighting conditions in aquatic environments. Additionally, comparative experiments were performed to evaluate the improvement in target detection accuracy by LMDNet under scenarios involving changes in water depth, target occlusion, and multiple targets.

4.4.1. Improvement in target detection accuracy by motion deblurring algorithms under different lighting conditions in aquatic environments

The motion-blurred images captured by the robotic tuna and the images processed using motion deblurring algorithms are shown in Fig. 12. The first column of Fig. 12 displays images under normal lighting conditions. As seen in the figure, compared to the blurred images, the deblurred images processed by the four motion deblurring algorithms, DeblurGAN-V2 (MobileNet), LMDNet, NAFNet-32, and LoFormer-S, show progressively clearer object boundaries, and motion blur artifacts gradually reduce. This demonstrates that using deblurring algorithms can restore richer details of the objects, which is beneficial for improving the accuracy of target detection algorithms. It also indicates that the performance of these four motion deblurring algorithms improves progressively. However, in the case of extreme motion blur, the deblurring algorithms can only partially recover from the blur. For example, the image in the first column of the first row in Fig. 12 is much clearer than the image in the first column of the sixth row. After deblurring, the results are displayed in the first column of rows 2 to 5 and rows 6 to 10, with the former being significantly clearer. The deblurring effect for extreme motion blur differs from that for mild motion blur, suggesting that the effectiveness of motion deblurring depends to some extent on the quality of the input image.

To discuss the effects of motion deblurring under different lighting conditions while maintaining the comparability of the experimental results, we used the Python image processing library PIL's ImageEnhance.Brightness function to adjust the brightness of the images under normal lighting conditions. The Brightness Factor variable was adjusted to modify the image brightness, with a factor of 0.7 reducing the brightness by 30 %, and a factor of 0.5 reducing the brightness by 50 %. The images processed with adjusted brightness are shown in the last four columns of the first and sixth rows in Fig. 12. As seen in the figure, even with decreased image brightness, the images processed by the four motion deblurring algorithms, DeblurGAN-V2 (MobileNet), LMD-

Net, NAFNet-32, and LoFormer-S, still achieved some degree of motion deblurring. However, under the darkest lighting condition in the fifth column, compared to the first and third columns, the deblurring effect is reduced. As the lighting dims, the features of the targets are further lost, thereby reducing the accuracy of target detection.

To further quantitatively demonstrate the improvement in target detection performance by motion deblurring algorithms under different lighting conditions, this study extracted frames from motion-blurred videos captured by the robotic tuna. A total of 3250 motion-blurred images, each containing a single robotic shark as the target, along with the corresponding brightness-adjusted images, were selected for comparison experiments. The experiments included detecting the 3250 motion-blurred images directly using the pre-trained YOLOv5m model with Brightness Factors of 1, 0.7, and 0.5, as well as detecting the images after applying motion deblurring algorithms (DeblurGAN-V2 (MobileNet), LMDNet, NAFNet-32, and LoFormer-S) and then using YOLOv5m for detection. The final experimental results are shown in Table 6. Under the same lighting conditions, all four algorithms, DeblurGAN-V2 (MobileNet), LMDNet, NAFNet-32, and LoFormer-S, effectively improved the target detection success rate of YOLOv5m, with the success rate increasing as the performance of the deblurring algorithm improved, as shown in Table 4. The motion deblurring performance follows the order: DeblurGAN-V2 (MobileNet) < LMDNet < NAFNet-32 < LoFormer-S. For example, when the Brightness Factor is 1, 0.7, and 0.5, the detection accuracy of YOLOv5m increased by 17.05 %, 13.39 %, and 10.00 %, respectively, when using LMDNet. These results show that the LMDNet motion deblurring algorithm effectively enhanced the performance of the YOLOv5m target detection model. The performance of the same motion deblurring algorithm is affected by the lighting intensity; as the lighting becomes dimmer, the improvement in target detection accuracy decreases. As shown in Tables 4 and 6, the motion deblurring effect of DeblurGAN-V2 (MobileNet) is inferior to LMDNet. Additionally, the computational cost of DeblurGAN-V2 (MobileNet) and LMDNet are 20.57 GMACs and 14.60 GMACs, respectively, with the computational cost of DeblurGAN-V2 (MobileNet) being 1.4 times that of LMDNet. Therefore, the overall performance of DeblurGAN-V2 (MobileNet) is not as good as LMDNet. Although NAFNet-32 and LoFormer-S have better motion deblurring performance compared to LMDNet, with target detection success rates improving by 19.51 % and 21.67 % when the Brightness Factor is 1, higher than the 17.05 % improvement of LMDNet, their computational costs are 86.76 GMACs and 221.01 GMACs, respectively, which are 5.9 times and 15.1 times higher than LMDNet.

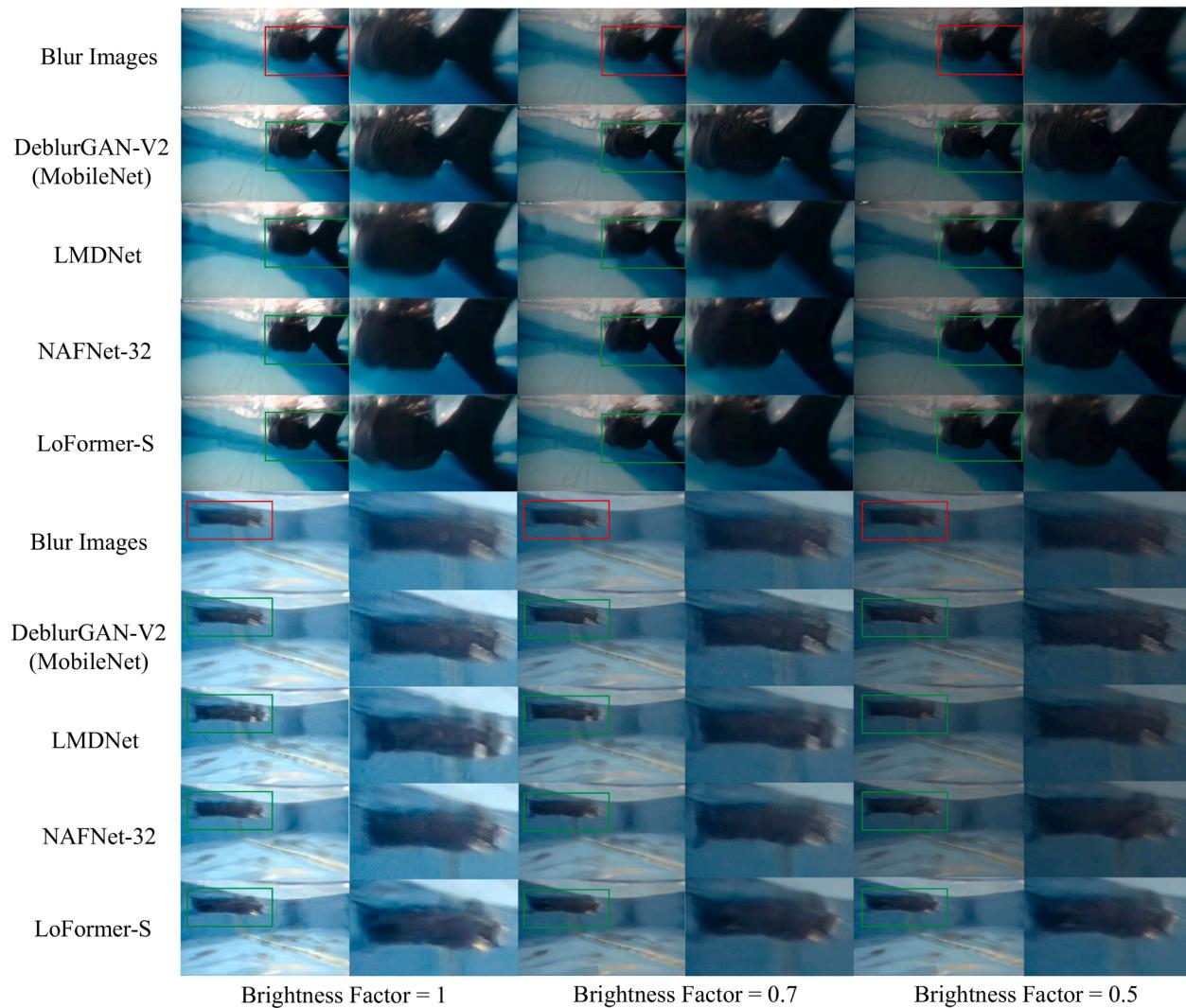


Fig. 12. Motion-blurred images captured by the robotic tuna and the deblurred images processed using motion deblurring algorithms under different lighting conditions. Columns 2, 4, and 6 are magnified views of the color boxes in columns 1, 3, and 5, respectively.

Table 6

Improvement in YOLOv5m detection success rate by different motion deblurring algorithms under various lighting conditions.

Brightness Factor	Algorithms	MACs of the motion deblurring algorithms (G)	Number of targets detected/Total number of targets	Detection success rate
1	YOLOv5m	-	2150/3250	66.15 %
	DeblurGAN-V2(MobileNet) + YOLOv5m	20.57	2424/3250	74.58 %(+ 8.43 %)
	LMDNet + YOLOv5m	14.60	2704/3250	83.20 %(+ 17.05 %)
	NAFNet-32 + YOLOv5m	86.76	2784/3250	85.66 %(+ 19.51 %)
	LoFormer-S + YOLOv5m	221.01	2854/3250	87.82 %(+ 21.67 %)
0.7	YOLOv5m	-	2095/3250	64.46 %
	DeblurGAN-V2(MobileNet) + YOLOv5m	20.57	2276/3250	70.03 %(+ 5.57 %)
	LMDNet + YOLOv5m	14.60	2530/3250	77.85 %(+ 13.39 %)
	NAFNet-32 + YOLOv5m	86.76	2647/3250	81.45 %(+ 16.99 %)
	LoFormer-S + YOLOv5m	221.01	2676/3250	82.34 %(+ 17.88 %)
0.5	YOLOv5m	-	1991/3250	61.26 %
	DeblurGAN-V2(MobileNet) + YOLOv5m	20.57	2143/3250	65.94 %(+ 4.68 %)
	LMDNet + YOLOv5m	14.60	2316/3250	71.26 %(+ 10.00 %)
	NAFNet-32 + YOLOv5m	86.76	2448/3250	75.32 %(+ 14.06 %)
	LoFormer-S + YOLOv5m	221.01	2508/3250	77.17 %(+ 15.91 %)

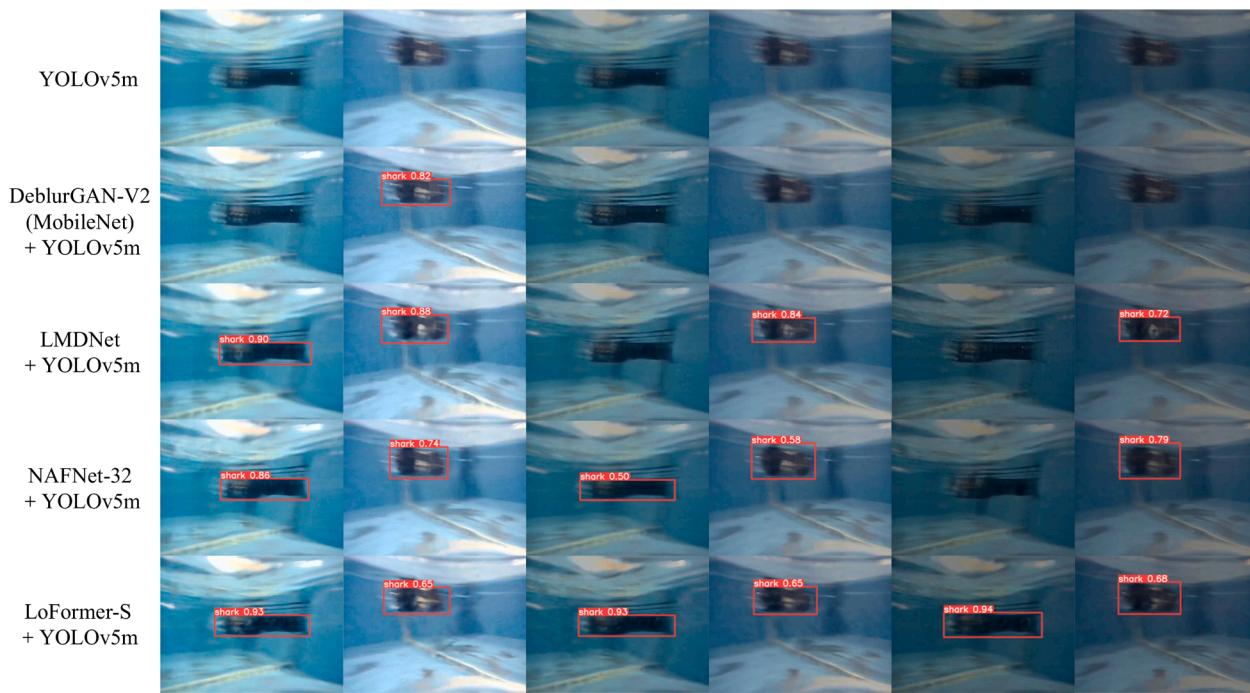


Fig. 13. Comparison of YOLOv5m Detection Results Using Different Motion Deblurring Algorithms. The first row shows motion-blurred images where YOLOv5m failed to detect the target, while the other rows display the YOLOv5m detection results after applying different motion deblurring algorithms.

Therefore, high-computation algorithms like NAFNet-32 and LoFormer-S are not suitable for edge computing deployment. Compared to DeblurGAN-V2 (MobileNet), NAFNet-32, and LoFormer-S, LMDNet achieves a good motion deblurring effect with the lowest computational cost, effectively balancing motion deblurring performance and computational cost. The target detection results of YOLOv5m using different motion deblurring algorithms are shown in Fig. 13. The deblurred images retain more detailed information, effectively improving the target detection success rate of YOLOv5m. The better the deblurring performance, the higher the improvement in the YOLOv5m target detection success rate. Meanwhile, as the lighting dims, the improvement in YOLOv5m's target detection success rate by the motion deblurring algorithms becomes more limited.

4.4.2. Improvement in target detection accuracy by LMDNet under water depth variation, target occlusion, and multi-target scenarios

To verify the effectiveness of LMDNet in improving target detection accuracy in more complex scenarios, while also conducting a quantitative analysis, this study extracted frames from motion-blurred videos captured by the robotic tuna. For each of the three scenarios, water depth variation, target occlusion, and multi-target scenes, 1800 motion-blurred images containing robotic sharks as the targets were selected. In the water depth variation scenario, the target's depth

changes between approximately 0.3m and 0.5m. In the target occlusion scenario, a blue plastic board was used as the occluder to randomly block the robotic shark target. In the multi-target scenario, three robotic sharks were used as targets. The corresponding scene images are shown in Fig. 14. The experimental results are shown in Table 7. In the water depth variation, target occlusion, and multi-target scenarios, LMDNet improved the YOLOv5m detection success rate by 15.89 %, 18.28 %, and 9.58 %, respectively. These results demonstrate that LMDNet can effectively improve target detection accuracy in these three scenarios.

In the water depth variation and target occlusion scenarios, LMDNet improved target detection accuracy by 15.89 % and 18.28 %, with no performance degradation. However, in the multi-target scenario, the improvement in target detection accuracy was only 9.58 %. The performance decline in this case is due to the fact that when multiple targets are present in the field of view, the targets need to be positioned farther from the robotic tuna to accommodate all the targets. In the target occlusion and multi-target scenarios, YOLOv5m's detection success rate also decreased due to the loss of target features and the greater distance of the targets. The YOLOv5m detection results using LMDNet in the water depth variation, target occlusion, and multi-target scenarios are shown in Fig. 14, where LMDNet effectively improves the detection success rate in all three scenarios.

Table 7
Improvement in YOLOv5m Detection Success Rate by LMDNet under Water Depth Variation, Target Occlusion, and Multi-Target Scenarios.

Experimental setups	Algorithms	Number of targets detected/ Total number of targets	Detection success rate
Water surface, non-occluded, single target	YOLOv5m	2150/3250	66.15 %
	LMDNet + YOLOv5m	2704/3250	83.20 % (+ 17.05 %)
Varying water depth	YOLOv5m	1175/1800	65.28 %
	LMDNet + YOLOv5m	1461/1800	81.17 % (+ 15.89 %)
Target occlusion	YOLOv5m	1024/1800	56.89 %
	LMDNet + YOLOv5m	1353/1800	75.17 % (+ 18.28 %)
Multiple target	YOLOv5m	2787/5400	51.61 %
	LMDNet + YOLOv5m	3304/5400	61.19 % (+ 9.58 %)

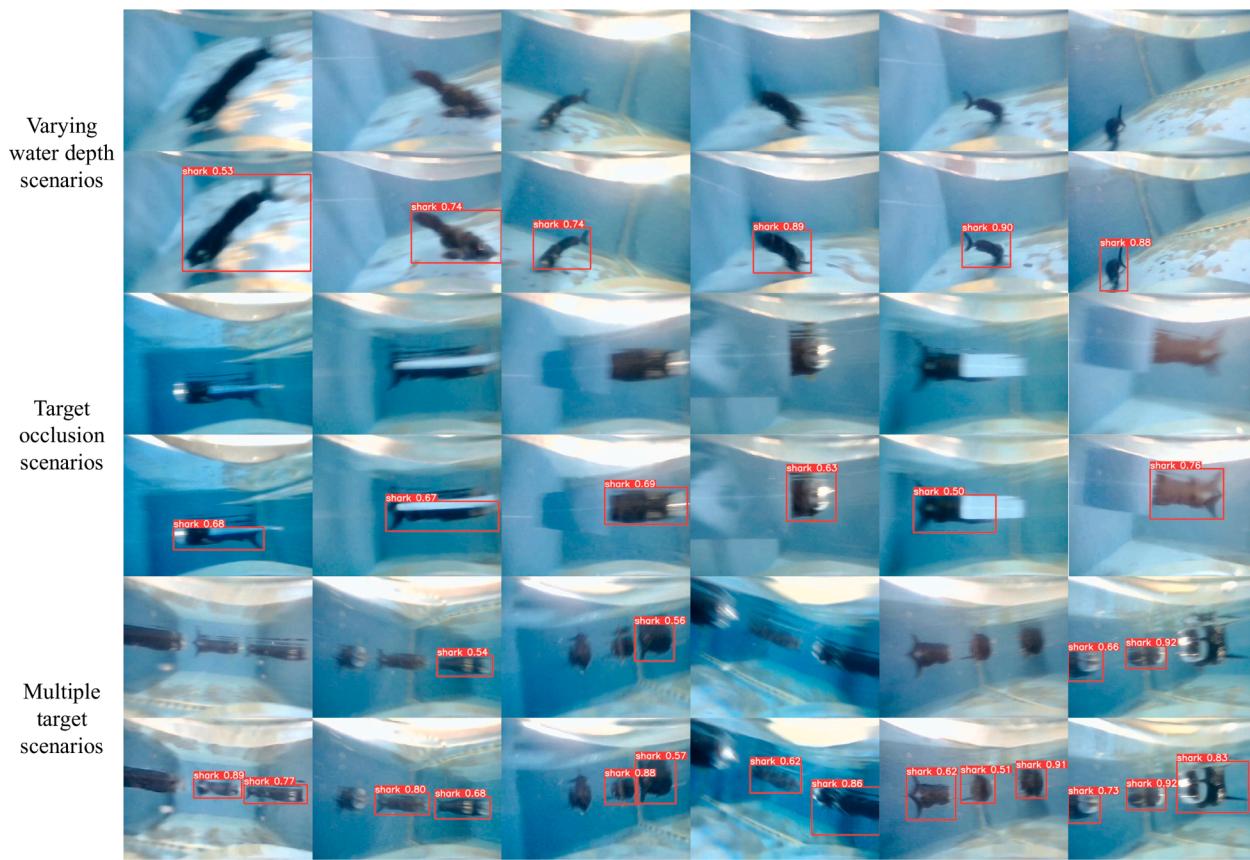


Fig. 14. Comparison of YOLOv5m Detection Results Using LMDNet under Water Depth Variation, Target Occlusion, and Multi-Target Scenarios. The first, third, and fifth rows show motion-blurred images where YOLOv5m detection was unsuccessful, while the second, fourth, and sixth rows show images where YOLOv5m detection was successful after LMDNet motion deblurring.

5. Implementation of autonomous tracking based on the LMDNet algorithm

5.1. Edge deployment of the algorithm and determination of target position information

The LMDNet and YOLOv5m models were jointly deployed on the Jetson Xavier NX, utilizing the camera mounted on the robotic fish to acquire video stream data. Considering the limited computational resources of the edge computing device, only one of the binocular camera's lenses was used to capture the video stream. The video stream has an image resolution of 640×360 . Since TensorRT can optimize deep learning models into inference engines for deployment on embedded platforms to accelerate inference, this study employed TensorRT for inference acceleration during model deployment. During the training and saving process, the parameters of both LMDNet and YOLOv5m were maintained at FP32 precision. Considering that quantization and pruning operations during edge computing deployment could lead to performance degradation, and to accelerate data flow processing in both models without compromising performance, we performed model conversion using TensorRT while retaining FP32 precision and avoiding quantization or pruning. The quantitative analysis of the LMDNet + YOLOv5m pipeline accelerated with TensorRT on the Jetson Xavier NX is shown in [Table 8](#). Without TensorRT acceleration, the pipeline processes each frame with a latency of 328ms and achieves a throughput of 3.05 FPS. With TensorRT acceleration on the Jetson Xavier NX, the latency per frame is reduced to 137ms, with a throughput of 7.30 FPS. This saves 191ms per image, making the processing speed 2.4 times faster than without TensorRT acceleration. In [Fig. 15](#), the images captured and processed by the camera during

Table 8

The quantitative analysis of the LMDNet + YOLOv5m pipeline accelerated with TensorRT on the Jetson Xavier.

Pipeline	TensorRT	Latency (ms)	Throughput (FPS)
LMDNet + YOLOv5m	✓	328	3.05
		137	7.30

the robotic fish target tracking experiment are shown. The outcomes of the experiment showed that over 117s, the joint operation of the LMDNet and YOLOv5m models processed a total of 852 frames, resulting in an average processing speed of 7.3 frames per second on the Jetson Xavier NX. The frequency at which target position information is generated is also 7.3 times per second, which can fulfill the requirements for control in real time and adjustment of the robotic fish.

Determining the 3D position of the target using a binocular camera and stereo matching algorithm consumes substantial computational resources on the edge computing device. Additionally, image distortion caused by the fish head casing needs to be calculated and corrected to obtain accurate position information. Therefore, this study abandoned the stereo matching method for position acquisition and instead used a fitting function, with the detected target's height h as the independent variable (since the target height only varies with the camera's distance and is not influenced by other factors) and the depth distance dis as the dependent variable. This fitting function approach incurs almost no additional computational cost while effectively and accurately estimating the distance between the camera and the robot shark.

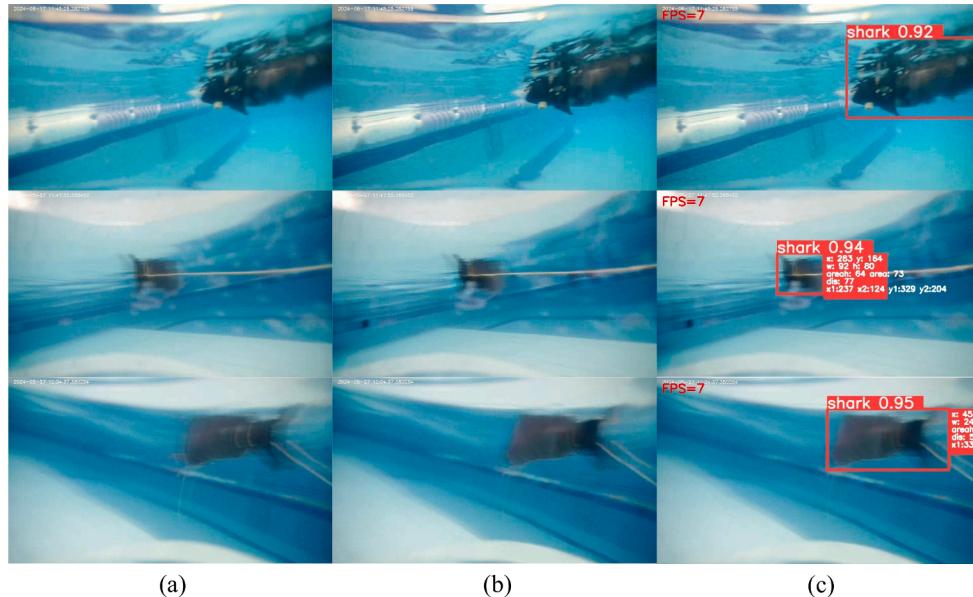


Fig. 15. Images captured by the camera during the robotic tuna target tracking experiment. Column a shows the original captured images, column b displays the LMDNet deblurred images, and column c presents the YOLOv5m target detection results on the deblurred images processed by LMDNet.

5.2. Robotic tuna autonomous visual target tracking control process

The robotic tuna tracking control experiment in this study was conducted following the flowchart shown in Fig. 16. During the control process, the horizontal coordinate difference between the detected target's position and the image center is denoted as Δx , while the angle between the vertical line through the fish head's centerline and the camera's centerline is represented as θ , as illustrated in Fig. 1. The set tracking distance is denoted as D_{track} , and the *Distance* represents the actual measured distance between the robotic tuna and the robotic shark. The difference between the tracking distance D_{track} and the actual distance *Distance* is defined as Δd . When the target is not found within the camera's field of view, the gimbal equipped with the camera will perform a back-and-forth sweep up to a maximum range of 270° to locate the robotic shark. Once the robotic shark is detected in the image, the robotic fish's waist-tail joint, pectoral fins, and propeller are controlled based on the target's position and depth to track the target, adjusting θ to approach 90° during the experiment. When the target is lost, the gimbal will resume its 270° sweeping motion to search for the target again.

5.3. Robotic tuna autonomous visual target tracking control experiment

An autonomous visual target tracking control experiment for the robotic tuna was conducted in a swimming pool, and a single com-

Table 9

Improvement of YOLOv5m detection performance using LMDNet in the robotic tuna target tracking experiment.

Algorithms	Number of targets detected/Total number of targets	Detection success rate
YOLOv5m	331/399	82.96 %
LMDNet + YOLOv5m	377/399	94.49 %

pleted experiment was selected for discussion. The experiment lasted for 57 s, during which 416 frames were processed. Out of these, 399 frames contained the robotic shark target. When using the combined LMDNet + YOLOv5m algorithm on the edge computing device, a total of 377 frames successfully detected the target. Our team preserved the original video stream data from the experiment, which represents the raw, unprocessed video stream. When using only the YOLOv5m algorithm, 331 frames successfully detected the target, as shown in Table 9. The detection success rate increased by 11.53 % with the deployment of LMDNet. This outcome provides additional proof of the motion deblurring algorithm's effectiveness in the robotic tuna's visual target tracking tasks. The improvement in target detection success rate will enhance the target perception ability of the robotic tuna during target tracking, reducing the probability of losing the target during the experiment.

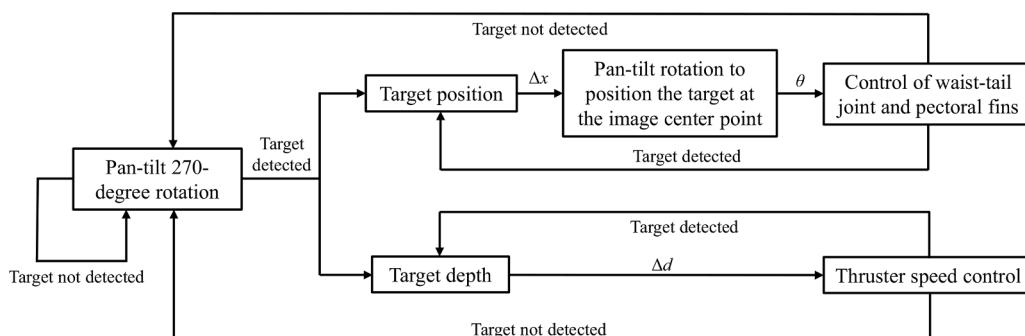


Fig. 16. Robotic tuna target tracking control flowchart.



Fig. 17. Sequence diagram of the robotic tuna tracking control experiment.

Fig. 17 presents the sequence diagram of the robotic tuna tracking the motion of the small robotic shark. In the subfigure on the left, the motion trajectory of the tracked small robotic shark is represented by the yellow path, while the red path is used to denote the motion trajectory of the robotic tuna. The similarity between the two trajectories suggests that autonomous target tracking control based on visual information was successfully implemented in this experiment. On the right side of **Fig. 17**, the sequence diagram of the robotic fish tracking process is shown, with a time interval of approximately 3 s between each subfigure.

Fig. 18 illustrates the variation of yaw angle, detection flag, θ value, and *Distance* (actual distance between the robotic fish) during the robotic tuna tracking control experiment, where θ represents the angle shown in **Fig. 1**. During the period from 0 to 16 s, the yaw angle of the robotic fish gradually decreased. Between 16 and 21 s, it changes from -180° to 180° , and a step change occurs around 20 s. From 21 to 62 s, the yaw angle gradually decreased from 180° to 0° , matching the changes in yaw angle shown in **Fig. 17**. The Detect Flag in **Fig. 18** represents the indicator of whether the robotic tuna has detected the target. It is evident that there is a target loss issue between 44 and 48 s. During this period, the robotic fish used the gimbal's 180-degree rotation to search for the target, causing the θ value to change linearly between 0 and 180° . The target was reacquired at 48 s. In the experiment, the tracking distance D_{track} was set to 70 cm. As shown in **Fig. 18**, the *Distance* value fluctuates around 70 cm. Therefore, this experiment not only validates the feasibility of the robotic tuna's target tracking control, but also demonstrates the effectiveness of the motion deblurring algorithm in enhancing the robotic fish's visual-based active tracking performance.

6. Discussion

6.1. Summary and discussion of algorithm experiments

Considering that multiple experiments on image deblurring and target detection have been conducted in this paper, in order to improve readability and facilitate summary and analysis, a cross-dataset comparison analysis table of motion deblurring and detection performance is presented, as shown in **Table 10**. Experiment 1 in **Table 10** corresponds to the experiment in **Section 4.2** of this paper. The improved LMDNet, based on the Baseline, achieves better motion deblurring performance with fewer additional parameters and computational cost. Experiment 2 in **Table 10** corresponds to the experiment in **Section 4.4.1**, where the detection success rate increased by 17.05% after motion deblurring using LMDNet. Experiment 3 in **Table 10** is conducted under the condition of a 50% reduction in lighting intensity, corresponding to the experiment in **Section 4.4.1**, where the detection success rate improved by 10.00%. Experiments 4, 5, and 6 in **Table 10** are conducted under different water depths, occluded targets, and multi-target scenarios, corresponding to the experiments in **Section 4.4.2**, where the detection success rates improved by 15.89%, 18.28%, and 9.58%, respectively. Experiment 7 in **Table 10** corresponds to the experiment in **Section 5.3**, where LMDNet and YOLOv5m were deployed for edge computing on the Jetson Xavier NX. The detection success rate improved by 11.53% after motion deblurring with LMDNet. The improvement in detection success rates from Experiment 2 to Experiment 7 demonstrates the effectiveness of LMDNet in motion deblurring across various scenarios. It also confirms that image deblurring benefits subsequent higher-level image

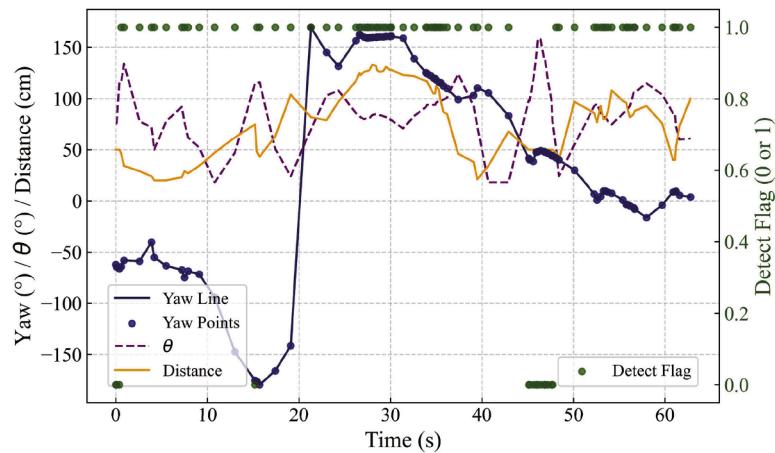


Fig. 18. Variation of yaw angle, detection flag, θ , and *Distance* (actual distance between the robotic fish) during the robotic tuna tracking control experiment.

Table 10

Cross-dataset comparison analysis of motion deblurring and detection performance.

Experiments	Experimental setups	Baseline	LMDNet	YOLOv5m	PSNR (GOPRO)	SSIM (GOPRO)	Successful detection targets/all targets	Detection success rate
1 (Section 4.2)	-	✓		✓	30.96 31.70	0.94 0.95	- -	- -
2 (Section 4.4.1)	Water surface, non-occluded, single target	✓	✓	✓	- -	- -	2150/3250 2704/3250	66.15 % 83.20 %
3 (Section 4.4.1)	Varying lighting (Brightness Factor 0.5)	✓	✓	✓	- -	- -	1991/3250 2316/3250	61.26 % 71.26 %
4 (Section 4.4.2)	Varying water depth	✓	✓	✓	- -	- -	1175/1800 1461/1800	65.28 % 81.17 %
5 (Section 4.4.2)	Target occlusion	✓	✓	✓	- -	- -	1024/1800 1353/1800	56.89 % 75.17 %
6 (Section 4.4.2)	Multiple target	✓	✓	✓	- -	- -	2787/5400 3304/5400	51.61 % 61.19 %
7 (Section 5.3)	Edge computing deployment	✓	✓	✓	- -	- -	331/399 377/399	82.96 % 94.49 %

processing. The reason why Experiment 2 shows a higher detection success rate than Experiment 7 is that the motion blur used in Experiment 2 is more severe, making it more challenging for the target detection algorithm. However, after motion deblurring with LMDNet, the images become more suitable for detection by the target detection algorithm. The improvement in detection success rates after LMDNet motion deblurring in Experiment 3 and Experiment 6 was smaller compared to Experiment 2, indicating that low-light conditions and multi-target scenarios lead to a decrease in LMDNet's motion deblurring performance.

6.2. Potential scalability of study

LMDNet has significant potential for scalability. For example, combining LMDNet with traditional motion deblurring algorithms could further improve deblurring performance while maintaining computational efficiency. Researching how LMDNet can be combined with brightness enhancement functions under low-light conditions to improve motion deblurring performance will help enhance the algorithm's robustness in low-light environments. By integrating data from other sensors, such as accelerometers, to achieve multimodal data fusion, the model could better understand the motion direction and speed of the image, which may optimize the motion blur effect. Researching LMDNet's adaptive capabilities, where the deblurring strategy can be automatically adjusted while ensuring lightweight operation, would enhance the algorithm's robustness in complex environments. Currently, LMDNet processes image data, but utilizing video stream data to capture correlations between consecutive frames could recover more details. Research into video sequence processing and motion deblurring based on stereo camera multi-source video stream data will improve stereo matching algorithms' localization accuracy.

LMDNet has the potential for application on other autonomous robotic platforms. This study has implemented the motion deblurring algorithm LMDNet for real-time image data processing on an underwater robotic fish edge computing platform, despite limited computational resources and physical space constraints. This demonstrates the feasibility of extending LMDNet to other autonomous robotic platforms. At the same time, as long as the edge computing devices on the autonomous robotic platform can support the lightweight model's operation, LMDNet's motion deblurring performance will not be compromised. Therefore, extending LMDNet to other autonomous robotic platforms for motion deblurring will be effective. Porting LMDNet to different autonomous robotic platforms is an engineering deployment issue. Specifically, the trained weight files of LMDNet can be converted to FP32 precision via TensorRT without modification, accelerating the model, and the generated model files can be deployed at the frontend

of the video stream processing on the autonomous robot platform to perform motion deblurring on the data flow. For example, in a drone equipped with object detection capabilities, if image motion blur occurs during autonomous flight, LMDNet can be deployed at the frontend of the object detection model. Before the data flow reaches the object detection model, it will first undergo motion deblurring processing by LMDNet. The output data stream from LMDNet will then be passed to the input of the object detection model, thereby achieving motion deblurring for the video stream.

The robotic fish tracking task also has potential scalability. In larger scenarios, the tracked target may be farther from the robotic fish, and during the real-time motion deblurring and target tracking process, the robotic fish may fail to locate the target's position information through motion deblurring combined with the object detection algorithm, ultimately leading to target loss. To address this issue, research can focus on the robotic fish's autonomous circling and cruising modes until the tracked target is successfully detected again, triggering the robotic fish's target tracking mode. In multi-target scenarios, studying how to track the same target is very meaningful. This can be achieved by adding a tracker in the detection model, allowing each target to have a unique ID number, and tracking the same target by following the target with the same ID number.

6.3. Limitations of the study

The work in this paper still has some limitations. For example, to meet the real-time requirements of motion deblurring, LMDNet must strike a balance between model size, computational complexity, and deblurring performance, which inevitably sacrifices some deblurring effectiveness. Although this study conducted experiments under different lighting conditions, varying depths, occluded targets, and multi-target scenarios, the task scenarios in this study still involve relatively simple underwater environments. LMDNet's deblurring performance is affected by lighting intensity, and under low-light conditions, the motion deblurring effect may degrade. As a common challenge in underwater optical image processing, LMDNet may not achieve good motion deblurring results in turbid underwater environments. As a lightweight motion deblurring method, LMDNet is suitable for edge computing devices. However, when processed on low-performance embedded devices with limited computational power and memory, it may exhibit significant performance differences. For devices with performance far below that of the Jetson Xavier NX, deployment personnel must consider the performance of the edge computing platform when deploying the model. LMDNet's training and evaluation were conducted on the public GoPro dataset, which may result in data shifts, and further efforts

are needed to improve the algorithm's generalization ability. Although LMDNet is a lightweight motion deblurring network, energy efficiency remains an important issue on some edge computing devices, and designing more energy-efficient models remains a challenge for the future. The lightweight deblurring model LMDNet performs well on relatively simple motion blur types, but it faces limitations in deblurring performance under extreme motion blur conditions. Therefore, LMDNet can only address motion blur to a certain extent.

7. Conclusion

This paper provides enhanced images for target detection in autonomous visual target tracking of bionic robotic fish and proposes a lightweight real-time motion deblurring algorithm, LMDNet, for image preprocessing. Firstly, a U-Net architecture, combined with the NAF-Block module from NAFNet, is utilized by LMDNet to construct its base network structure. Subsequently, the TLC method is applied for global feature aggregation to mitigate performance degradation caused by feature distribution shifts. Additionally, the Ghost module is integrated to create more feature maps at a reduced computational cost, thereby enhancing feature extraction. Finally, a wavelet loss function tailored for motion deblurring is proposed. Experimental results show that LMDNet is a lightweight model that can be deployed on edge computing devices, with the number of parameters and computational cost being only 2.7M and 14.6 GMACs, respectively. The network achieves a PSNR of 31.70 on the public GOPRO dataset, outperforming other lightweight motion deblurring algorithms. Although LMDNet's PSNR and SSIM scores are slightly lower compared to state-of-the-art motion deblurring algorithms, it possesses significant application potential due to its lower computational cost and edge computing capabilities. Additionally, LMDNet improves the detection success rate of the YOLOv5m target detection model by 17.05 % on the motion-blurred dataset used in this study, validating the effectiveness of LMDNet in enhancing target detection performance.

Additionally, the joint deployment and operation of the LMDNet motion deblurring algorithm and the YOLOv5m target detection algorithm were successfully achieved on the Jetson Xavier NX in this study. The effectiveness of the motion deblurring algorithm was further validated by the experimental results. A visual perception mechanism for target tracking experiments was designed in this study, with the small robotic shark as the target. Using the target position information obtained by the integrated algorithm, the robotic tuna was controlled accordingly. The autonomous visual target tracking experiment of the robotic tuna, based on visual motion deblurring and target detection, was successfully completed, and the expected results were achieved. The future research directions of this paper include exploring more efficient lightweight motion deblurring models that further improve deblurring performance while maintaining real-time capabilities; studying how to enhance the algorithm's robustness in varying environments, such as low light, complex environmental noise, and real underwater scenarios; improving the model's cross-dataset robustness through adaptive mechanisms and transfer learning; developing more robust robotic tuna tracking control methods; and implementing the functionality of tracking specific targets in scenarios involving multiple targets.

Data availability

Data will be made available on request.

CRediT authorship contribution statement

Yang Liu: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization; **Bingxiong Wang:** Methodology, Formal analysis; **Runtong Ai:** Methodology, Formal analysis; **Guohua Yu:**

Methodology, Formal analysis; **Yinjie Ren:** Methodology, Formal analysis; **Jincun Liu:** Conceptualization, Methodology, Investigation, Resources, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition; **Yaoguang Wei:** Conceptualization, Supervision, Project administration, Funding acquisition; **Dong An:** Conceptualization, Supervision, Project administration, Funding acquisition; **Yan Meng:** Conceptualization, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant 62273351 and grant 62303020.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at [10.1016/j.eswa.2025.128326](https://doi.org/10.1016/j.eswa.2025.128326)

References

- Bahat, Y., Efrat, N., & Irani, M. (2017). Non-uniform blind deblurring by reblurring. In *Proceedings of the IEEE international conference on computer vision* (pp. 3286–3294).
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint [arXiv:2004.10934](https://arxiv.org/abs/2004.10934).
- Chen, D., Wu, Z., Meng, Y., Tan, M., & Yu, J. (2022a). Development of a high-speed swimming robot with the capability of fish-like leaping. *IEEE/ASME Transactions on Mechatronics*, 27(5), 3579–3589.
- Chen, G., Yang, X., Xu, Y., Lu, Y., & Hu, H. (2022b). Neural network-based motion modeling and control of water-actuated soft robotic fish. *Smart Materials and Structures*, 32(1), 015004.
- Chen, L., Chu, X., Zhang, X., & Sun, J. (2022c). Simple baselines for image restoration. In *European conference on computer vision* (pp. 17–33). Springer.
- Chu, X., Chen, L., Chen, C., & Lu, X. (2022). Improving image restoration by revisiting global information aggregation. In *European conference on computer vision* (pp. 53–71). Springer.
- Cotter, F. (2020). Uses of complex wavelets in deep convolutional neural networks. Doctoral dissertation UNIVERSITY OF CAMBRIDGE.
- Cui, Y., Ren, W., Cao, X., & Knoll, A. (2024). Image restoration via frequency selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(2), 1093–1108.
- Cui, Z., Li, L., Wang, Y., Zhong, Z., & Li, J. (2023). Review of research and control technology of underwater bionic robots. *Intelligent Marine Technology and Systems*, 1(1), 7.
- Fang, Z., Wu, F., Dong, W., Li, X., Wu, J., & Shi, G. (2023). Self-supervised non-uniform kernel estimation with flow-based motion prior for blind image deblurring. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 18105–18114).
- Gao, H., Yang, J., Zhang, Y., Wang, N., Yang, J., & Dang, D. (2023). A mountain-shaped single-stage network for accurate image restoration. arXiv preprint.
- Ghasemabadi, A., Janjua, M. K., Salameh, M., Zhou, C., Sun, F., & Niu, D. (2024). Cascaded-edge gaze: Efficiency in global context extraction for image restoration. arXiv preprint [arXiv:2401.15235](https://arxiv.org/abs/2401.15235).
- Gu, A., & Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint [arXiv:2312.00752](https://arxiv.org/abs/2312.00752).
- Guo, C., Wang, Q., Dai, H.-N., Wang, H., & Li, P. (2022). Lnnet: Lightweight nested network for motion deblurring. *Journal of Systems Architecture*, 129, 102584.
- Han, J., Guo, J., Zhao, Z., Ma, Z., Liu, Y., & Zou, S. (2023). Orientation control and target tracking for bionic robotic fish based on ADRC. In *2023 42nd Chinese control conference (CCC)* (pp. 4137–4142). IEEE.
- Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., & Xu, C. (2020). Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1580–1589).
- Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132–7141).
- Hu, Y., Zhao, W., Wang, L., & Jia, Y. (2009). Underwater target following with a vision-based autonomous robotic fish. In *2009 American control conference* (pp. 5265–5270). IEEE.
- Ji, D., Rehman, F. U., Ajwad, S. A., Shahani, K., Sharma, S., Sutton, R., Li, S., Ye, Z., Zhu, H., & Zhu, S. (2020). Design and development of autonomous robotic fish for object detection and tracking. *International Journal of Advanced Robotic Systems*, 17(3), 1729881420925284.

- Ji, Y., Wei, Y., Liu, J., & An, D. (2023). Design and realization of a novel hybrid-drive robotic fish for aquaculture water quality monitoring. *Journal of Bionic Engineering*, 20(2), 543–557.
- Kim, Y., Kwon, H., & Ko, H. (2024). Facial image deblurring network for robust illumination adaptation and key structure restoration. *Engineering Applications of Artificial Intelligence*, 133, 107959.
- Kong, L., Dong, J., Ge, J., Li, M., & Pan, J. (2023). Efficient frequency domain-based transformers for high-quality image deblurring. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5886–5895).
- Kupyn, O., Martyniuk, T., Wu, J., & Wang, Z. (2019). Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 8878–8887).
- Liu, J., Zhang, C., Liu, Z., Zhao, R., An, D., Wei, Y., Wu, Z., & Yu, J. (2021). Design and analysis of a novel tendon-driven continuum robotic dolphin. *Bioinspiration & Biomimetics*, 16(6), 065002.
- Liu, S., Wang, Y., Li, Z., Jin, M., Ren, L., & Liu, C. (2022a). A fluid-driven soft robotic fish inspired by fish muscle architecture. *Bioinspiration & Biomimetics*, 17(2), 026009.
- Liu, Y., Anderlini, E., Wang, S., Ma, S., & Ding, Z. (2022b). Ocean explorations using autonomy: Technologies, strategies and applications. In *Offshore robotics: Volume I issue 1, 2021* (pp. 35–58). Springer.
- Liu, Y., Liu, Z., Yang, H., Liu, Z., & Liu, J. (2023). Design and realization of a novel robotic manta ray for sea cucumber recognition, location, and approach. *Biomimetics*, 8(4), 345.
- Mao, X., Li, Q., & Wang, Y. (2024a). Adarevd: Adaptive patch exiting reversible decoder pushes the limit of image deblurring. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 25681–25690).
- Mao, X., Wang, J., Xie, X., Li, Q., & Wang, Y. (2024b). Loformer: Local frequency transformer for image deblurring. In *Proceedings of the 32nd ACM international conference on multimedia* (pp. 10382–10391).
- Meng, Y., Wu, Z., Dong, H., Wang, J., & Yu, J. (2020). Toward a novel robotic manta with unique pectoral fins. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(3), 1663–1673.
- Meng, Y., Wu, Z., Li, Y., Chen, D., Tan, M., & Yu, J. (2023). Vision-based underwater target following control of an agile robotic manta with flexible pectoral fins. *IEEE Robotics and Automation Letters*, 8(4), 2277–2284.
- Meng, Y., Wu, Z., Zhang, P., Wang, J., & Yu, J. (2022). Real-time digital video stabilization of bioinspired robotic fish using estimation-and-prediction framework. *IEEE/ASME Transactions on Mechatronics*, 27(6), 4281–4292.
- Nah, S., Hyun Kim, T., & Mu Lee, K. (2017). Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3883–3891).
- Qiu, C., Wu, Z., Wang, J., Tan, M., & Yu, J. (2022). Locomotion optimization of a tendon-driven robotic fish with variable passive tail fin. *IEEE Transactions on Industrial Electronics*, 70(5), 4983–4992.
- Qiu, C., Wu, Z., Wang, J., Tan, M., & Yu, J. (2023). Autonomous recovery control of biomimetic robotic fish based on multi-sensory system. *IEEE Robotics and Automation Letters*, 9(2), 1420–1427.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788).
- Redmon, J., & Farhadi, A. (2017). Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263–7271).
- Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention-MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18* (pp. 234–241). Springer.
- Schmidt, U., Rother, C., Nowozin, S., Jancsary, J., & Roth, S. (2013). Discriminative non-blind deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 604–611).
- Sharif, S., Naqvi, R. A., Ali, F., & Biswas, M. (2023). Darkdeblur: Learning single-shot image deblurring in low-light condition. *Expert Systems with Applications*, 222, 119739.
- Sun, F., Yu, J., Chen, S., & Xu, D. (2014). Active visual tracking of free-swimming robotic fish based on automatic recognition. In *Proceeding of the 11th world congress on intelligent control and automation* (pp. 2879–2884). IEEE.
- Sun, Q., Han, J., Li, H., Liu, S., Shen, S., Zhang, Y., & Sheng, J. (2020). A miniature robotic turtle with target tracking and wireless charging systems based on IPMCs. *IEEE Access*, 8, 187156–187164.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., & Li, H. (2022a). Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 17683–17693).
- Wang, Z., Dong, C., Zhang, Z., Tian, Q., Sun, A., Yuan, L., & Zhang, L. (2022b). Review of multi-fin propulsion and functional materials of underwater bionic robotic fish. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 236(13), 7350–7367.
- Xu, L., & Jia, J. (2010). Two-phase kernel estimation for robust motion deblurring. In *Computer vision-ECCV 2010: 11th European conference on computer vision, Heraklion, Crete, Greece, September 5–11, 2010, proceedings, part I 11* (pp. 157–170). Springer.
- Xu, L., Tao, X., & Jia, J. (2014). Inverse kernels for fast spatial deconvolution. In *Computer vision-ECCV 2014: 13th European conference, Zurich, Switzerland, September 6–12, 2014, proceedings, part V 13* (pp. 33–48). Springer.
- Yan, S., Wang, J., Wu, Z., Tan, M., & Yu, J. (2023). Autonomous vision-based navigation and stability augmentation control of a biomimetic robotic Hammerhead Shark. *IEEE Transactions on Automation Science and Engineering*, 21(3), 3319–3331.
- Yang, X., Wu, Z., & Yu, J. (2016). A novel active tracking system for robotic fish based on cascade control structure. In *2016 IEEE international conference on robotics and biomimetics (ROBIO)* (pp. 749–754). IEEE.
- Yu, J., Sun, F., Xu, D., & Tan, M. (2015). Embedded vision-guided 3-d tracking control for robotic fish. *IEEE Transactions on Industrial Electronics*, 63(1), 355–363.
- Yu, J., Wu, Z., Yang, X., Yang, Y., & Zhang, P. (2020). Underwater target tracking control of an untethered robotic fish with a camera stabilizer. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(10), 6523–6534.
- Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., Yang, M.-H., & Shao, L. (2021). Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14821–14831).
- Zhang, B., Sun, J., Sun, F., Wang, F., & Zhu, B. (2024). Image deblurring method based on self-attention and residual wavelet transform. *Expert Systems with Applications*, 244, 123005.
- Zhang, K., Ren, W., Luo, W., Lai, W.-S., Stenger, B., Yang, M.-H., & Li, H. (2022). Deep image deblurring: A survey. *International Journal of Computer Vision*, 130(9), 2103–2130.