IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 April 2024, accepted 6 May 2024, date of publication 9 May 2024, date of current version 16 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3399077

RESEARCH ARTICLE

An Adapted ResNet-50 Architecture for Predicting Flow Fields of an Underwater Vehicle

ZIFENG DUAN[1], FANG WANG[1], BIAO WANG[1], GAOSHENG LUO[1], AND ZHE JIANG[1,2]

[1]Shanghai Engineering Research Center of Hadal Science and Technology, College of Engineering Science and Technology, Shanghai Ocean University, Shanghai 201306, China

[2]Shanghai Jiao Tong University Underwater Engineering Institute Company Ltd., Shanghai 200231, China

Corresponding author: Zhe Jiang (zjiang@shou.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFC2808105.

ABSTRACT This study introduces FlowRes, an adapted ResNet-50 architecture, to predict flow fields around underwater vehicles, aiming to improve the efficiency of Computational Fluid Dynamics (CFD) through deep learning. The background highlights the necessity for rapid and accurate flow field predictions to enhance the hydrodynamic shape design of an Autonomous Remote-Controlled Vehicle (ARV) for inspection of offshore energy underwater infrastructure. Employing a decoder-only CNN-based model, the methodology involves modifying ResNet-50 for image-to-image generation, focusing on flow field visualization of underwater vehicles using a compact dataset. Results from training the model with 240 paired entries of flow fields and design parameters show significant computational speed improvements, with generated images deviating less than 1% from traditional CFD-generated images. The conclusions underline the potential of integrating advanced deep learning techniques with CFD, demonstrating FlowRes's capability in significantly accelerating the design process of underwater vehicles by offering a faster, more resource-efficient alternative to traditional methods.

INDEX TERMS Computational fluid dynamics, deep learning, flow field prediction, underwater vehicles design.

I. INTRODUCTION

Computational Fluid Dynamics (CFD) is a pivotal tool in simulating fluid behavior by solving the Navier-Stokes equations [1], yielding valuable data like scalars and imagery for research. Typically, simulating flow fields around objects involves modeling, mesh generation, computation, and visualization. However, this process can be time-consuming and expensive, influenced by factors such as computational cell count, solution methods, and additional variables.

Deep learning improves upon traditional CFD by offering significant computational speed and efficiency, enabling faster predictions, including real-time analysis in some instances, exceling in handling complex, high-dimensional problems with a level of computational inexpensiveness that CFD cannot match. This makes deep learning a valuable tool for scenarios requiring rapid results or the analysis of multiple conditions, thereby enhancing the overall efficiency of fluid dynamics research and application.

Deep learning research in flow field prediction bifurcates into two domains: one focusing on variable flow conditions [2], [3], [4], [5] and the other on shape design variables [6], [7], [8], [9]. Notably, there has been substantial research on the former. Guo et al. [10] were the first to introduce convolutional neural network (CNN) [11], [12] for flow field prediction, focusing specifically on generating real-time velocity fields for non-uniform steady laminar flows. Following this foundational work, subsequent researchers have built upon and expanded these techniques. Bhatnagar et al.'s research [13] employed a CNN-based network to examine the effects of attack angles and Reynolds numbers on airfoil flow fields. This work utilized

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

a shared encoder for computational efficiency and trained separate decoders. Kashir et al. [14] introduced an innovative architecture transforming CFD vertices into point clouds, demonstrating accuracy and computational agility with an airfoil test case.

In contrast, research on shape parameter-based flow field prediction remains scarce, mainly due to the complexities introduced by meshing alterations and solver preferences. Duru et al.'s CNNFOIL [9] estimates airfoil pressure fields across various shapes, achieving notable accuracy and potential for generalization. Wu et al. developed ffsGAN [6] and daGAN [8], GAN-structured models, with daGAN integrating a CNN to address gaps in supercritical airfoil research with CFD wind-tunnel experiments. Jiang et al.'s TransCFD [7] utilized a transformer architecture [15] for predicting airfoil velocity and pressure flow fields. These studies collectively underscore the potential and capabilities of artificial intelligence, particularly deep learning, in the preliminary phases of airfoil design. They are significant in exploring design parameters, offering valuable insights in a time-efficient manner. The real-time level feedback on flow fields provided by these AI-driven methods is not only meaningful but also reduces the time and resource investment typically required in traditional design processes.

While encoder-decoder architectures [16] offer versatile output generation, they can be cumbersome for precise predictions. Hence, a decoder-only model [17] may be more suitable for our study. Notably, the application of AI has been largely confined to aerodynamics and 2D airfoils. Our work with FlowRes, a decoder-only CNN-based model, seeks to expand the application of CNNs for early-stage underwater vehicle design, focusing on flow field prediction.

The research focuses on a developing Autonomous Remote-Controlled Vehicle (ARV) [18] for the offshore energy sector, addressing the critical need for efficient and cost-effective cable inspection and maintenance. Offshore oil platforms and wind power platforms are reliant on various seabed cables for essential functions, including power supply, control of underwater pipeline valves, chemical injection, production monitoring, and multifunctional operations. The safety and functionality of these platforms are highly dependent on the integrity of these cables.

For offshore oil platforms, these cables are vital for connecting platforms, controlling wellhead devices, and integrating various operations [19]. Similarly, in offshore wind energy [20], cables are crucial for connecting wind turbines to the grid. Regular inspection and maintenance of these cables are essential, especially in the context of wind farms, where high-voltage alternating current (HVAC) sea cables can transmit over distances of up to 70 kilometers.

Traditional methods of cable inspection are often expensive and inefficient. To address this, the ARV involved is designed to autonomously locate, track, and inspect sea cables over long periods. It will be capable of identifying issues like cable displacement, exposure, and damage, and can also be manually operated for inspection and basic tasks. The ARV will be adaptable, and able to be equipped with different end-effectors for tasks such as cutting, moving cables, and removing debris like entangled fishing nets. This innovation is aligned with the evolving maintenance needs and industry trends in offshore energy infrastructure, offering a more cost-effective and efficient solution for cable maintenance.

A primary objective in cable inspecting and maintaining ARV design research is finding the most efficient parameter combinations within the design space. Traditional CFD tools linearly escalate the time investment with case scale, especially when simulating variations of a single prototype with similar flow patterns. Our research introduces FlowRes, a novel approach employing a ResNet-50-based decoder [21] to supplant the conventional meshing, computation, and visualization stages. FlowRes aims to delineate the relationship between design parameters and flow fields using a compact dataset, potentially reducing computational demands for simulating various cases significantly.

The paper is structured as follows. Section I, introduction, lays the foundation for the study by highlighting the development tendency of integration of deep learning techniques with CFD to address the challenges faced by traditional methods. Section II, FlowRes Architecture, details the adapting ResNet-50 for fluid dynamics, focusing on model modifications for image-to-image predictions. Section III, Experiment Setup, outlines the modeling process, including the prototype selection and CFD configuration for underwater vehicles. Section IV, Results and Discussion, compares FlowRes model predictions with traditional CFD results and discusses design optimization for drag reduction. Finally, Section V, concludes the paper, summarizing key findings and proposing future work directions.

II. FLOWRES ARCHITECTURE

A. RESNET-50 INTRODUCTION

ResNet-50, standing for Residual Network with 50 layers (shown in Figure 1), is a groundbreaking architecture in the realm of deep learning, particularly within computer vision. ResNet-50 was conceived out of a necessity to overcome the limitations imposed by the vanishing gradient problem in very deep neural networks. This problem, which results in the gradient becoming increasingly smaller as it is propagated back through the network, makes it difficult for deep networks to learn effectively.

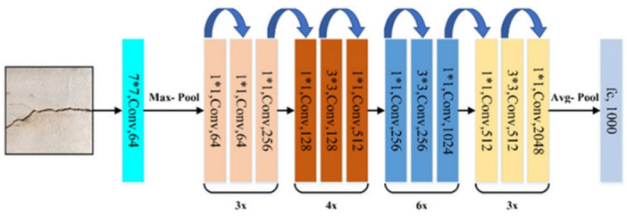

FIGURE 1. ResNet-50 diagram.

The primary mission ResNet-50 was designed to accomplish was to significantly improve the performance on visual recognition tasks, such as image classification, by enabling the training of much deeper neural networks without suffering from the vanishing gradient issue. Its innovation lies in its ability to train such deep networks while maintaining or even improving accuracy, a challenge that was previously thought to be insurmountable for conventional network architectures.

The core innovation of ResNet-50 is the introduction of the residual block (shown in Figure 2). This architectural feature allows the network to skip one or more layers through shortcut connections that perform identity mapping, with no additional parameters or computational complexity. These shortcuts enable the direct propagation of the gradient back through the network, mitigating the vanishing gradient problem and allowing for the effective training of deeper networks. This approach contrasts with traditional architectures, where each layer needs to learn the desired underlying mapping directly.

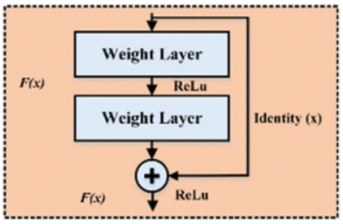

FIGURE 2. Residual block diagram.

ResNet-50's groundbreaking approach to deep learning, characterized by residual learning and shortcut connections, has transcended its original computer vision domain to influence a wide array of fields. These include natural language processing [22], [23], audio recognition [24], medical image analysis [25], and even the prediction of flow fields [3], [4], [6], [9], [14], [23]. By enabling the effective training of very deep neural networks, ResNet-50 has significantly enhanced performance across diverse sequence-to-sequence tasks. Its architecture has spurred further research and innovation, leading to the development of more sophisticated and efficient neural network models. Consequently, ResNet-50 stands as a key milestone in deep learning, driving advancements not only in computer vision but also across various domains where deep learning finds application.

B. NEURAL NETWORK ARCHITECTURE

FlowRes, modified from the standard ResNet-50, is designed to predict CFD flow field images of 128 × 128 pixels in RGB channels. Due to its adaptation for distinct tasks, several adjustments were necessary. Firstly, the probability prediction output of ResNet-50 is substituted with sequence output, manifesting as a complete picture. Additionally, the absence of gradual up-sampling of feature maps during processing in ResNet-50 necessitated its incorporation into FlowRes. Significant alterations in FlowRes compared to ResNet-50 include:

Removal of the conv1 block, including the 7 × 7 convolution layer, max pooling layers, and the final average pooling layer with softmax activation.

Adjustment of the first convolution layers in each block to a stride of 1.

Addition of a linear embedding layer at the beginning to encode initial parameters.

Inclusion of a 1 × 1 convolution layer and a sigmoid activation function at the end for image output.

These modifications transform the standard ResNet-50 from a classification-oriented model to an image-to-image generative model. In FlowRes, the process decreases and merges channels while increasing the feature map size, with channel changes achieved through binary interpolation.

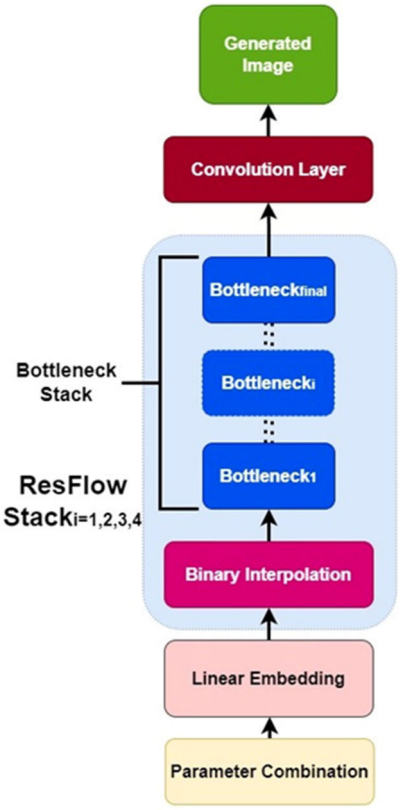

FIGURE 3. FlowRes architecture.

The main architecture of FlowRes, depicted in the Figure 3, begins by encoding the combination of 8 design parameters into a latent space (H0×W0×C, in this case, 8×8×128) using a linear layer. This latent space represents a C-dimensional set of H0×W0 size feature maps. The data then passes through four FlowRes blocks, each containing a binary interpolation operation that doubles the resolution of the feature maps. Each FlowRes block includes identical bottlenecks, shown in Figure 4, that extract and process features, reducing dimensions by half. The feature map set, after processing,

enters a 1 × 1 convolution layer with a sigmoid activation function, culminating in the final 128 × 128×3 image output.

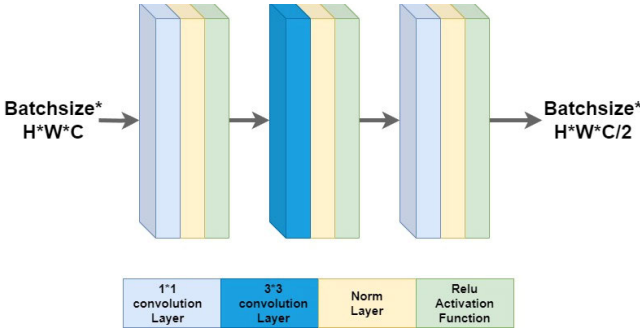

FIGURE 4. Bottleneck structure.

III. MODEL SETUP

A. MODELING

Our dataset comprises parameter combinations and corresponding CFD result pairs, derived from a prototype underwater vehicle. To improve simulation efficiency, the vehicle model was simplified by removing appendages, filling holes, and smoothing gaps, as depicted in Figure 5, 6. The initial underwater vehicle model and its simplified model are acquired by SOLIDWORKS.

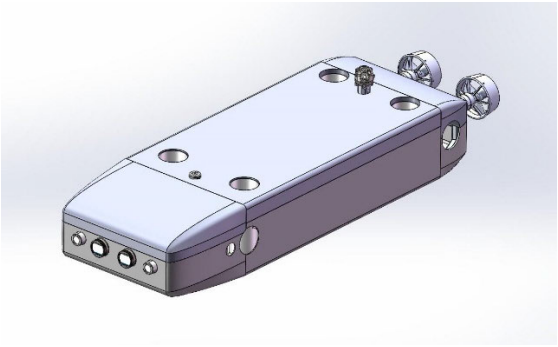

FIGURE 5. Underwater vehicle prototype.

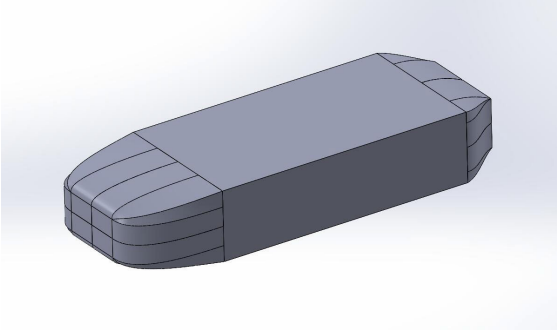

FIGURE 6. Simplified model.

The outline of the simplified model's shape is meticulously defined by a set of formulae [26], specifying the contours and dimensions of the design. These formulae delineate precise lines, which form the basis of the model's structure. Subsequently, the models are generated through a methodical process: the lines defined by these formulae are first created, followed by the development of a mesh surface that spans across these established lines. This methodical approach ensures that the simplified models are not only accurate in their representation but also consistent with the theoretical design parameters set forth in our study. For a visual representation of these simplified models and an overview of the associated variables, refer to the diagrams provided in Figure 7 and Figure 8.

FIGURE 7. Simplified model side view.

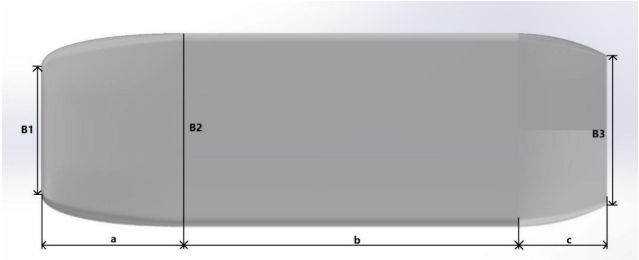

FIGURE 8. Simplified model top view.

Bow section longitudinal profile curve equation:

$$r1-1(x)=\frac{H1}{2}+\frac{H2-H1}{2}[1-(1-\frac{x}{a})^{n1}]^{\frac{1}{m1}} \quad (1)$$

Bow section transverse profile curve equation:

$$r1-2(x)=\frac{B1}{2}+\frac{B2-B1}{2}[1-(1-\frac{x}{a})^{n2}]^{\frac{1}{m2}} \quad (2)$$

Parallel mid-body section longitudinal profile curve equation:

$$r2-1(x)=\frac{H2}{2} \quad (3)$$

Parallel mid-body section transverse profile curve equation:

$$r2-2(x)=\frac{B2}{2} \quad (4)$$

Stern section longitudinal profile curve equation:

$$r3-1(x)=\frac{H2}{2}-[\frac{3(H2-H3)}{2c^2}-\frac{\tan\theta 1}{c}](x-a-b)^2+[\frac{H2-H3}{c^3}-\frac{\tan\theta 1}{c^2}](x-a-b)^3 \quad (5)$$

TABLE 1. Table of variable description.

Symbol	Quantity
a	Bow distance
b	Parallel mid-body distance
c	Stern distance
B_1	Bow front width
B_2	Parallel mid-body width
B_3	Stern tail width
H_1	Bow front height
H_2	Parallel mid-body height
H_3	Stern tail height
m_1	Fullness coefficient 1 of the bow longitudinal curve
n_1	Fullness coefficient 2 of the bow longitudinal curve
m_2	Fullness coefficient 1 of the bow transverse curve
n_2	Fullness coefficient 2 of the bow transverse curve
$tan\theta_1$	Tangent of tail semi-angle 1
$tan\theta_2$	Tangent of tail semi-angle 2
x	Distance from bow center point
$r_{1\text{-}1}(x)$	Bow height function depending on x
$r_{1\text{-}2}(x)$	Bow width function depending on x
$r_{2\text{-}1}(x)$	Mid-body height function depending on x

Stern section transverse profile curve equation:

$$r3-2(x) = \frac{B2}{2} - [\frac{3(B2-B3)}{2c^2} - \frac{\tan\theta 2}{c}](x-a-b)^2 + [\frac{B2-B3}{c^3} - \frac{\tan\theta 2}{c^2}](x-a-b)^3 \quad (6)$$

The descriptions of each variable are shown in Table 1,

B. SAMPLING

After acquiring the initial simplified model, sampling is conducted to obtain a parameter library for the further study. The sampling strategy involves generating sampling points with values ranging from 0 to 1 on each dimension, which are then mapped independently to the corresponding variable ranges in the optimization space.

Optimal Latin Hypercube Sampling (OLHS) is an advanced variant of the Latin Hypercube Sampling (LHS) method [27], aimed at enhancing the efficiency and effectiveness of sampling in multidimensional parameter spaces. While LHS ensures an even exploration across the entire range of each parameter by dividing its distribution into equally probable intervals, OLHS goes a step further by optimizing the selection of sample points within these intervals. The optimization criteria often involve minimizing the correlation between parameters or maximizing the minimum distance between points, thus ensuring a more evenly distributed and representative sample. However, A common bias is the inadvertent introduction of sampling bias, where certain regions of the input space are overrepresented or underrepresented due to the sampling strategy employed. This can lead to skewed model performance and compromised generalizability, particularly if the dataset does not adequately capture the true distribution of the underlying data.

The adoption of OLHS as the sampling method in our paper is justified by its several advantages over traditional LHS, particularly when implemented using Python 3 to achieve the maximization of minimal distance. This criterion significantly improves the representativeness of the sample by optimizing the spatial distribution, effectively minimizing clustering and gaps among sample points. Consequently, this leads to more accurate and stable estimates of statistical moments and model responses. Moreover, the enhanced independence of simulations due to reduced correlation between sampled points is crucial for effective uncertainty and sensitivity analyses. Additionally, OLHS's ability to provide a more uniform coverage of the parameter space improves the efficiency of computational resources, especially valuable in complex models requiring a large number of simulations. These advantages make OLHS a superior choice for exploring multidimensional parameter spaces with high accuracy and efficiency.

C. SENSITIVITY ANALYSIS

A comprehensive sensitivity analysis was undertaken to identify an optimal CFD configuration. This analysis encompassed several parameters: the dimensions of the computational domain, the quantity of the computational mesh cells, and the choice of turbulence model. The common conditions applied across all experimental scenarios are delineated in Table 2.

TABLE 2. CFD condition table.

Condition Type	Selection
Pressure and velocity coupling method	SIMPLE algorithm
Momentum equation difference scheme	second-order upwind
Turbulence equation difference scheme	second-order upwind
Inlet condition	velocity inlet/velocity is 2.057m/s
Outlet condition	pressure outlet/pressure is 0

In the initial experiment, the computational domain was a cuboid extending three times the vehicle's length ($3L$) to the rear and 1.5 times the length ($1.5L$) to the front, with distance in other directions as variable, detailed in Table 3.

TABLE 3. Computational domain sensitivity analysis result.

Distance	Pressure Force(N)	Change ratio
$0.5L$	108.82	
$1L$	99.13	8.90%
$1.5L$	97.36	1.79%
$2L$	96.3	1.09%
$2.5L$	95.95	0.36%

Analysis revealed that at a distance of $1.5L$, the variation in pressure force did not exceed 2% from the preceding data

point, indicating convergence. The computational domain of this distance is shown in the Figure 9.

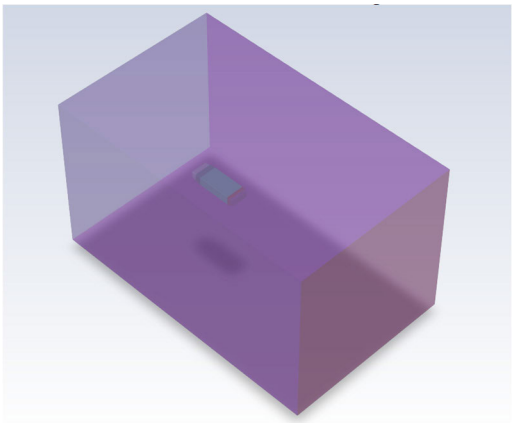

FIGURE 9. Computational Domain of 1.5L Distance.

The second experiment established that a mesh quantity of 5.3 million mesh cells resulted in pressure force variation within 2% of the previous measure, suggesting a threshold for computational accuracy, detailed in Table 4. The demonstration of the 5.3 million mesh cells configuration is shown in the Figure 10.

TABLE 4. Quantity of the computational mesh cells sensitivity analysis result.

Case Number	Mesh Cell Quantity	Pressure Force(N)	Change ratio
1	0.2k	148.54	
2	1.01M	101.46	31.70%
3	2.03M	99.56	1.87%
4	3.56M	97.33	2.24%
5	5.30M	96.9	0.44%

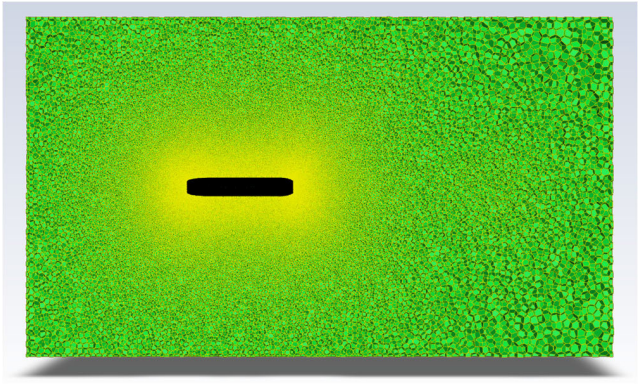

FIGURE 10. Computational Domain of Case 5.

The final experiment evaluated the performance of different turbulence models—$k-\epsilon$ *standard*, *SST* $k-\omega$, and *RSM* under uniform conditions to ascertain their accuracy in simulating flow fields. The resulting flow fields are illustrated in Figure, highlighting the comparative performance of each turbulence model in capturing the complex vortex dynamics.

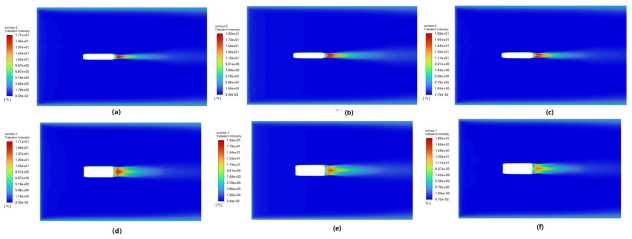

FIGURE 11. Turbulence Pressure Contours of Vehicle under Different Turbulence Models (a) k-epsilon standard model side-view turbulence pressure contour (b) k-omega SST model side-view turbulence pressure contour (c) RSM model side-view turbulence pressure contour (d) k-epsilon standard model top-view turbulence pressure contour (e) k-omega SST model top-view turbulence pressure contour(f) RSM model top-view turbulence pressure contour.

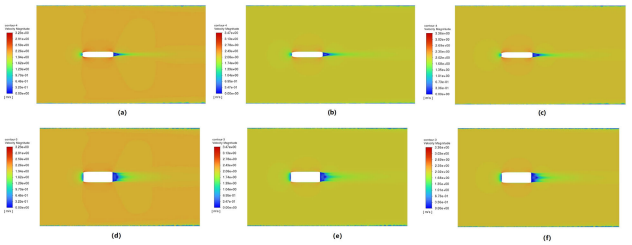

FIGURE 12. Velocity Contours of Vehicle under Different Turbulence Models (a) k-epsilon standard model side-view velocity contour (b) k-omega SST model side-view velocity contour (c) RSM model side-view velocity contour (d) k-epsilon standard model top-view velocity contour (e) k-omega SST model top-view velocity contour (f) RSM model top-view velocity contour.

The analysis focused on vortex pressure and distribution at the vehicle's stern, where three distinct vortices were observed. The $k-\epsilon$ *standard* model predicted lower vertical separation and higher horizontal pressure of vortex clusters compared to *RSM*, with *SST* $k-\omega$ results intermediate. Velocity field analysis revealed zero-speed regions at both ends of the submersible, indicative of flow separation, with variations consistent across models. The $k-\epsilon$ *standard* model, in particular, provided an accurate simulation of flow patterns and pressure in the stern low-pressure zone, aligning with empirical expectations. Given its suitability for bluff body characteristics and moderate computational demand, the $k-\epsilon$ *standard* model was selected for further simulation in the study.

From the conducted experiments, the optimal CFD setup conditions identified for subsequent research include a computational domain extending $3L$ behind and $1.5L$ in other directions of the vehicle, with a mesh quantity of 5.3 million mesh cells, using the $k-\epsilon$ *standard* turbulence model. These conditions ensure computational efficiency while maintaining accuracy in simulating flow patterns, particularly in capturing the flow separation and vortex formation around the vehicle. This configuration provides a robust foundation for further investigations.

IV. RESULTS AND DISCUSSION

A. WORKFLOW

The experimental workflow is illustrated in Figure 13, segmented into four distinct phases. The library of ARV parameters is constructed using the OLHS method, as outlined in the experimental setup. The parameters utilized, defining the bow shapes, are comprehensively outlined in Table 5. The internal space of the underwater vehicle bow is dictated by the onboard instruments, including 1 pressure shell, 3 thrusters, 2 underwater lights, and 2 underwater cameras. Notably, the width of the underwater lights (101mm) and the width of the underwater cameras (85mm) influence parameter B_1, while the volume of the remaining devices sets the lower limit for other parameters. Subsequently, modeling of the ARV based on this library is conducted in SOLIDWORKS, utilizing a Visual Basic for Applications script for automation. Further automation is achieved through a Python script, which commands Fluent to perform calculations on the models under the specified experimental conditions. The visual outcomes of these calculations are compiled into a library of flow fields, represented in JPEG format.

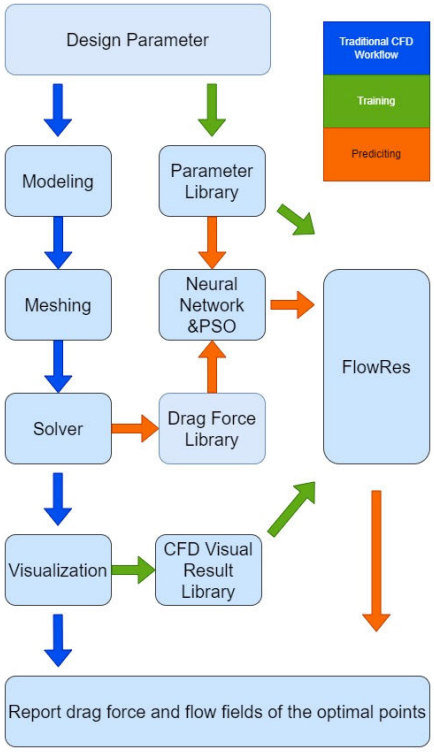

FIGURE 13. Experimental workflow.

Subsequently, the parameter library and CFD results dataset serve as inputs, with the corresponding flow field targets, to facilitate the training of the FlowRes model. Throughout the training phase, we distinctly categorized the images into two types: pressure field images and velocity field images. The validation dataset is then employed to evaluate the predictive accuracy of the model. In the final stage, a neural network is utilized to elucidate the relationship between the parameter library and the drag force library. Particle Swarm Optimization (PSO) [28] is subsequently applied to identify optimal points characterized by minimal drag force. The validated FlowRes model is then used to predict the flow fields associated with these optimal points.

TABLE 5. CFD condition table.

Corresponding Control Variable	Range
B_1(mm)	[400,800]
m_1	[1.5,3.0]
m_2	[1.5,3.0]
n_1	[1.5,3.0]
n_2	[1.5,3.0]

In our experiment, the objective was to train the FlowRes model to accurately generate flow field images corresponding to a variety of tow shapes of underwater vehicles. To achieve our objective, we carefully compiled a dataset containing 240 paired entries, each linking an image depicting the flow field to its corresponding design parameters. To ensure robust validation and guard against overfitting, we implemented a 5-fold cross-validation technique. This involved systematically dividing the dataset into five subsets, with each fold consisting of a distinct validation set comprising 20% of the data (48 pairs), while the remaining 80% (192 pairs) were earmarked for the training set. The approach enabled thorough model evaluation and exploration of varied data patterns across multiple validation folds, thereby enhancing the reliability and generalization capability of our deep learning model.

B. EVALUATION METRICS

Mean Squared Error (MSE) stands as a commonly used metric in statistical analysis and machine learning, offering a quantitative measure to assess the average squared differences between predicted values and actual observations. Particularly valuable in evaluating regression models aimed at predicting continuous numeric values, MSE provides a reliable indicator of model accuracy. The training process was guided by MSE metric, a critical measure used both for guiding the training and for the final evaluation of the model. The calculations are as below,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2 \tag{7}$$

$\widehat{Y}_i$, Y_i and N denote predicted result and the CFD result and total number of pixels. This metric provides a quantitative way to assess the accuracy of the model, where a lower MSE value indicates a better fit between the model's predictions and the actual data.

1) TRAINING SETTINGS

For the training process, we employed the Adam optimizer with a constant learning rate of 0.001. We set the batch size to 16 and trained the model over 300 epochs. The resulting average MSE values for the generated images, acquired by using the validation set, were 0.00638 for pressure and 0.00682 for velocity. The implementation, training, and prediction of all experiments are done based on PyTorch 2.2 and a standard Ubuntu-20.04 OS with an Intel Xeon Platinum 8380 CPU.

C. DECODER COMPARISON

We utilized ViTGAN and SAGAN, two state-of-the-art models, to individually construct decoders. FlowRes, along with these two decoders, underwent parameter adjustments to standardize their predictive performance. The results are presented in Table 6. FlowRes exhibited the lowest MSE in its prediction outputs. We attribute this superiority to its operational methodology. While Generative Adversarial Networks (GANs) excel in numerous generative tasks, they necessitate extensive datasets for robust training. Given our relatively small dataset, GANs typically do not undergo thorough training. Furthermore, GANs are more suited for tasks requiring diverse outputs, whereas our objective demands precision and singularity. Additionally, in the context of flow field prediction, transformers often perform excessive global computations. This occurs because minor variations in physical parameters occur in regions distant from the vehicle. Our focus lies in capturing detailed changes proximal to the vehicle.

TABLE 6. Decoders performance comparison.

Decoder	MSE
FlowRes	0.0064
SAGAN	0.0084
ViTGAN	0.0095

D. OPTIMIZATION

In the study, we further enhanced our methodology by incorporating PSO. This technique was used to strategically identify design parameters that minimize pressure drag force for the underwater vehicles. Utilizing the trained neural network, which focused on the relationship between shape design parameters and pressure drag force, PSO helped pinpoint three optimal design points, shown in Table 7.

Finally, the optimal point was processed through both the traditional Computational CFD workflow and the FlowRes workflow to generate visualization results, depicted in Figure 14. The outcomes of each workflow were then compared. The images of pressure and velocity fields produced by FlowRes demonstrated less than 1% deviation from those generated by the traditional CFD method, as determined by MSE. Impressively, the time required for predicting a case was reduced from 25 minutes to just a few seconds, significantly enhancing efficiency over the traditional CFD workflow. The results underscored the utility of our integrated approach in the early stages of underwater vehicle design, presenting a time-efficient method for optimizing design parameters.

TABLE 7. Optimal point.

Variable	Optimal point
B_1(mm)	619.5254
m_1	2.572784
m_2	2.404145
n_1	2.317325
n_2	2.135482
Pressure Drag Force(N)	95.18

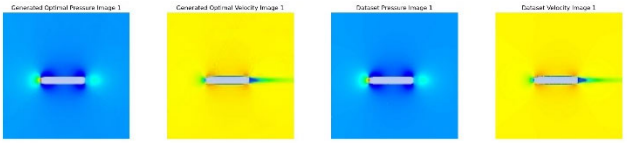

FIGURE 14. Optimal points comparison between FlowRes and CFD results.

V. CONCLUSION

In this study, we introduce FlowRes, which shows novelty on two sides: first, through its utilization of the ResNet-50 architecture for flow field prediction, and second, by its adaptation specifically for ARV early design phases. This innovative approach integrates deep learning into the initial stages of ARV design, and showcases a notable improvement in simulation efficiency. In addition, it is essential to note that while this study introduces FlowRes, an application of the ResNet-50 architecture tailored for use in the early design phases of ARV development, the full extent of its validity and efficiency in other tasks within the realm of hydrodynamics remains to be thoroughly investigated. FlowRes represents a novel approach by integrating deep learning methodologies into the initial stages of ARV design, showcasing its potential compared to the traditional simulation techniques. However, further research and experimentation, such as testing the performance of FlowRes in the design of other types of underwater vehicles, are warranted to explore the broader applicability and effectiveness of FlowRes across various hydrodynamic tasks associated with flow field prediction.

Despite the promising findings attributed to FlowRes, certain limitations persist, warranting further investigation. Firstly, the efficacy of the data-driven approach is contingent upon the quality of the dataset, with our validation efforts limited to assessing the decoder's precision solely on a simplified ARV model. However, in the context of more intricate appli-

cations involving underwater vehicles, there arises a pressing need for more expansive datasets capable of capturing the intricacies of detail-rich environments. Secondly, both the resolution of the flow field predictions and the training dataset utilized are low-resolution. To enhance the applicability of FlowRes in practical engineering scenarios, an upgrade in resolution is imperative. Nonetheless, this endeavor presents notable challenges, particularly concerning the architectural design of FlowRes and its computational capabilities.

By leveraging the capabilities of FlowRes, we have observed an acceleration in identifying flow fields, achieving speeds up to four orders of magnitude faster than traditional CFD methods. This remarkable enhancement in simulation efficiency can be attributed to a fundamental distinction between CFD and neural networks. While traditional CFD methods require iterative calculations for each case, accumulating computation time with each iteration, FlowRes undergoes an initial training phase on a dataset. Consequently, when presented with a new case for prediction, FlowRes requires only a single run based on the previous training, resulting in significantly reduced computation time. This improvement is not merely incremental; it represents a significant leap forward in the efficiency of simulations, offering a more rapid and resource-efficient process for the early design phases of underwater vehicles.

The application of FlowRes in ARV design is a testament to the potential of integrating advanced deep learning techniques, such as ResNet, with established CFD principles. This synergy not only enhances the predictive accuracy and speed of simulations but also paves the way for more innovative approaches in the field. Serving as the inaugural application of ResNet in the ARV early design, FlowRes contributes to the progression of realistic engineering needs, furnishing a sturdy and effective instrument for the initial design of ARV. With its nimble and immediate flow field forecasting capability, the duration of the design iteration can be significantly curtailed. Upon successful training and implementation of FlowRes, designers reap substantial time savings in comparison to conventional CFD workflows, thereby expediting the entire design process.

ACKNOWLEDGMENT

The authors extend their appreciation to Dr. Li Xin of Shanghai Ocean University, for supplying the high-performance computing (HPC) resources, which are essential for their CFD calculations and neural network training phases.

REFERENCES

[1] R. Temam, *Navier-Stokes Equations: Theory and Numerical Analysis*. Providence, RI, USA: AMS Chelsea Publishing, 2001.
[2] F. D. A. Belbute-Peres, T. Economon, and Z. Kolter, "Combining differentiable PDE solvers and graph neural networks for fluid flow prediction," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 2402–2411.
[3] X. Hui, J. Bai, H. Wang, and Y. Zhang, "Fast pressure distribution prediction of airfoils using deep learning," *Aerosp. Sci. Technol.*, vol. 105, Oct. 2020, Art. no. 105949.
[4] M. D. Ribeiro, A. Rehman, S. Ahmed, and A. Dengel, "DeepCFD: Efficient steady-state laminar flow approximation with deep convolutional neural networks," 2020, *arXiv:2004.08826*.
[5] V. Sekar, M. Zhang, C. Shu, and B. C. Khoo, "Inverse design of airfoil using a deep convolutional neural network," *AIAA J.*, vol. 57, no. 3, pp. 993–1003, Mar. 2019.
[6] H. Wu, X. Liu, W. An, S. Chen, and H. Lyu, "A deep learning approach for efficiently and accurately evaluating the flow field of supercritical airfoils," *Comput. Fluids*, vol. 198, Feb. 2020, Art. no. 104393.
[7] J. Jiang, G. Li, Y. Jiang, L. Zhang, and X. Deng, "TransCFD: A transformer-based decoder for flow field prediction," *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106340.
[8] H. Wu, X. Liu, W. An, and H. Lyu, "A generative deep learning framework for airfoil flow field prediction with sparse data," *Chin. J. Aeronaut.*, vol. 35, no. 1, pp. 470–484, Jan. 2022.
[9] C. Duru, H. Alemdar, and Ö. U. Baran, "CNNFOIL: Convolutional encoder decoder modeling for pressure fields around airfoils," *Neural Comput. Appl.*, vol. 33, no. 12, pp. 6835–6849, Jun. 2021.
[10] X. Guo, W. Li, and F. Iorio, "Convolutional neural networks for steady flow approximation," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 481–490.
[11] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, Apr. 1980.
[12] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.
[13] S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik, "Prediction of aerodynamic flow fields using convolutional neural networks," *Comput. Mech.*, vol. 64, no. 2, pp. 525–545, Aug. 2019.
[14] B. Kashir, M. Ragone, A. Ramasubramanian, V. Yurkiv, and F. Mashayek, "Application of fully convolutional neural networks for feature extraction in fluid flow," *J. Visualizat.*, vol. 24, no. 4, pp. 771–785, Aug. 2021.
[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 2–8.
[16] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–4.
[17] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, San Francisco, CA, USA, Tech. Rep., 2018.
[18] Y. R. Petillot, G. Antonelli, G. Casalino, and F. Ferreira, "Underwater robots: From remotely operated vehicles to intervention-autonomous underwater vehicles," *IEEE Robot. Autom. Mag.*, vol. 26, no. 2, pp. 94–101, Jun. 2019.
[19] W. F. Nielsen and T. Davenport, "Submarine cables and offshore energy," in *Submarine Cables*. Leiden, The Netherlands: Brill Nijhoff, 2014.
[20] E. Pan, B. Yue, X. Li, Z. Zhao, and Q. Zhu, "Integration technology and practice for long-distance offshore wind power in China," *Energy Convers. Econ.*, vol. 1, no. 1, pp. 4–19, Sep. 2020.
[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[22] J. Wang, Y. Xu, J. Hu, M. Yan, J. Sang, and Q. Qian, "Improved visual fine-tuning with natural language supervision," 2023, *arXiv:2304.01489*.
[23] S. Wang, B. Wang, J. Gong, Z. Wang, X. Hu, X. Duan, Z. Shen, G. Yue, R. Fu, D. Wu, W. Che, S. Wang, G. Hu, and T. Liu, "Combining ResNet and transformer for Chinese grammatical error diagnosis," in *Proc. 6th Workshop Natural Lang. Process. Techn. Educ. Appl.*, 2020, pp. 36–43.
[24] T. Zhou, Y. Zhao, and J. Wu, "ResNeXt and Res2Net structures for speaker verification," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Jan. 2021, pp. 301–307.
[25] L. Shen, L. R. Margolies, J. H. Rothstein, E. Fluder, R. McBride, and W. Sieh, "Deep learning to improve breast cancer detection on screening mammography," *Sci. Rep.*, vol. 9, no. 1, p. 12495, Aug. 2019.
[26] D. F. Myring, "A theoretical study of body drag in subcritical axisymmetric flow," *Aeronaut. Quart.*, vol. 27, no. 3, pp. 186–194, Aug. 1976.
[27] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, p. 239, May 1979.
[28] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," *Chemometric Intell. Lab. Syst.*, vol. 149, pp. 153–165, Dec. 2015.

ZIFENG DUAN is currently pursuing the master's degree in naval architecture and ocean engineering with Shanghai Ocean University. His research interests include the application of AI in fluid dynamics and hydrodynamic analysis.

FANG WANG received the Ph.D. degree from the School of Naval Architecture, Civil and Ocean Engineering, Shanghai Jiao Tong University, in 2007. She did postdoctoral research with Pusan National University, South Korea, from 2007 to 2008. She worked on underwater structures and materials with China Ship Scientific Research Center, from 2008 to 2013, and she has been engaged in structure safety assessment of marine structures with Shanghai Ocean University, since 2013. Her research interests include material selection, strength analysis, fatigue, and fracture performance of marine structures. Meanwhile, she is an Associate Editor of *Ocean Engineering* and a member of the International Ship and Offshore Structures Congress.

BIAO WANG received the Ph.D. degree in naval architecture and ocean engineering from Shanghai Jiao Tong University, in 2015. He is currently an Assistant Research Fellow with the College of Marine Science, Shanghai Ocean University. His research interests include unmanned underwater vehicles, marine trenchers, intelligent control systems, and autonomous systems.

GAOSHENG LUO received the Ph.D. degree in mechanical electronic engineering from Zhejiang University, in 2013. He is currently an Assistant Research Fellow with the College of Marine Science, Shanghai Ocean University. His research interests include the development and control of full-ocean-depth manipulators and underwater robotics and utilizing advanced robotics operating systems.

ZHE JIANG was born in October 1982. He received the Ph.D. degree. He is currently the Deputy Director of Shanghai Engineering Research Center of Hadal Science and Technology. He is a Professor. He has been engaged in the design of deepsea offshore platforms, the design of deep-sea submersibles and intelligent underwater robots for subsea structures, and hydrodynamics, for many years. His current research interests include the intelligent underwater robots for the operation and maintenance of offshore platforms and wind turbines, deepsea in-situ observation and exploration equipment, submersible hydrodynamics, and safety assessment based on real-time inspection data and artificial intelligence method.