



**HAL**  
open science

# An Automatic Highly Dynamical Digital Twin Design with YOLOv8 for hydrodynamic studies on living animals

Bastien Lagneaux, Gurvan Jodin, Dixia Fan, James Herbert-Read, Corentin  
Porcon, Florence Razan

► **To cite this version:**

Bastien Lagneaux, Gurvan Jodin, Dixia Fan, James Herbert-Read, Corentin Porcon, et al.. An Automatic Highly Dynamical Digital Twin Design with YOLOv8 for hydrodynamic studies on living animals. 2024 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA), Feb 2024, Victoria, Seychelles. 10.1109/acdsa59508.2024.10467329 . hal-04527017

**HAL Id: hal-04527017**

**<https://hal.science/hal-04527017>**

Submitted on 30 Apr 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# An Automatic Highly Dynamical Digital Twin Design with YOLOv8 for hydrodynamic studies on living animals.

Bastien Lagneaux  
*dept. of Mecatronics*  
IETR UMR CNRS 6164, ENS Rennes  
Bruz, France  
bastien.lagneaux@ens-rennes.fr

Gurvan Jodin  
*dept. of Mecatronics*  
IETR UMR CNRS 6164, ENS Rennes  
Bruz, France  
gurvan.jodin@ens-rennes.fr

Dixia Fan  
*school of Engineering*  
Westlake University  
Hangzhou, China  
fandixia@westlake.edu.cn

James Herbert-Read  
*dept. of Zoology*  
Cambridge University  
Cambridge, United Kingdom  
jh2223@cam.ac.uk

Corentin Porcon  
*dept. of Mecatronics*  
IETR UMR CNRS 6164, ENS Rennes  
Bruz, France  
corentin.porcon@ens-rennes.fr

Florence Razan  
*dept. of Mecatronics*  
OASIS, IETR UMR CNRS 6164  
University of Rennes, Rennes, France  
florence.razan@ens-rennes.fr

**Abstract**—This paper presents an innovative method using the YOLOv8 algorithm to automate the creation of digital twins (DTs) replicating hydrodynamic behaviors of real organisms in liquid environments. The approach extracts features from video data, facilitating efficient DT generation. Addressing computational challenges in accurately simulating fish movements, our method offers insights into animals' perception and use of hydrodynamic cues. Training YOLO on a custom dataset, processing predictions, integrating with computational fluid dynamics (CFD), and comparative analysis against ground truth simulations demonstrate the effectiveness of our automated digital twin creation.

**Index Terms**—Computer Vision, Artificial Intelligence, Automation, Virtual Engineering

## I. INTRODUCTION

In this study, we propose a groundbreaking approach to automating the creation of digital twins (DTs) that faithfully represent the hydrodynamic characteristics of real organisms, particularly in liquid environments. Leveraging the YOLOv8 algorithm [1], we aim to extract pertinent features from video data, offering a swift and cost-effective means to generate DTs. The automation of DT creation is essential to manage the extensive volume of experiments conducted by biologists, facilitating robust statistical analyses. This innovative methodology addresses the computational challenges associated with constructing a DT capable of accurately simulating the intricate movements of real fish, opening new avenues for gaining profound insights into how animals perceive and leverage hydrodynamic cues.

This project received funding from the Human Frontier Science Program Organization under grant no. RGY0059/2022.

Indeed, in the realm of aquatic environments, animals leave behind hydrodynamic motion cues—a distinctive 'liquid fingerprint'—containing crucial details about size, identity, and past movements [2]. This study takes a pivotal role in a larger project involving the development of a sophisticated analytical tool. This tool combines data from physical sensors with the aforementioned swift and cost-effective simulation approach. The overarching goal is to comprehensively decode hydrodynamic cues and understand how animals interpret and utilize them in liquid environments. Fish, in particular, exhibit sensitivity to these cues, generating vortices with pressure, velocity, and vorticity information detectable by canal neuromasts [3]–[5]. Despite the known role of these cues in optimizing swimming efficiency [6], their full potential for guiding animals in search and navigational avoidance tasks remains largely unexplored.

In the subsequent sections, we will outline our approach, detailing the processing of predictions of a YOLOv8 model, trained on a custom dataset, the integration into computational fluid dynamics (CFD), and the comparative analysis against ground truth simulations. Through this comprehensive approach, we aim to demonstrate the effectiveness of our automated digital twin creation method in capturing dynamic hydrodynamic behaviors of real organisms.

## II. STATE OF THE ART

The immense potential of YOLO [7] and other Computer Vision algorithms in image analysis has been repeatedly demonstrated, with numerous studies leveraging their capabilities to varying extents in the development of digital twins.

While YOLO is a powerful tool, it is not the sole option, as demonstrated in [8]. This study builds a digital twin using OpenCV, recognizing the diversity of tools available. The emphasis here is on simplicity and precision, showcasing the versatility of computer vision techniques beyond YOLO. However, for our work, this solution is unsuitable due to color similarities between certain fish parts and their shadows, hence the choice to delve into YOLO’s possibilities.

The fundamental application of YOLO indeed lies in image analysis, making it particularly useful to scrutinize images resulting from CFD. This concept is succinctly captured in [9], which highlights YOLO’s use for automatic flow structure detection and tracking in shadowgraph imaging within experimental fluid dynamics. This paves the way for harnessing YOLO’s capabilities to analyze flow fields using high-speed cameras.

Moving beyond its fundamental application, YOLO proves highly beneficial for modifying the digital twin of a low dynamic system. [10]–[12] collectively demonstrate this versatility. In [10], YOLOv5 is introduced for object detection in a Building Digital Twin (BDT), showcasing its potential in creating dynamic digital twins within a building environment. [11] addresses the necessity for high-quality geometric models in digital twins, combining dynamic mode decomposition (DMD) for motion detection, YOLOv5 for object detection, and 3D machine learning for pose estimation. The focus here is on detecting changes in rotational poses to minimize storage and bandwidth requirements. In [12], YOLOv7-X is employed to construct a virtual apple orchard, proposing a semantic communication framework that reduces data transmission costs through object detection and introduces resource allocation schemes for enhanced transmission quality.

Moreover, YOLO finds application in constructing more dynamic digital twins, especially in scenarios where objects exhibit rapid movement, as showcased in [13]. This study employs a hybrid deep neural network comprising MobileNetv2, YOLOv4, and Openpose for small object detection in a Digital Twin (DT) for smart manufacturing. The objective is to achieve dynamic synchronization between the physical manufacturing system and its virtual representation.

While the applications discussed herein provide dynamic representations of digital twins by focusing on shape and positioning, [14] underscores a critical aspect. It emphasizes that achieving enhanced fidelity in digital twins necessitates a shift towards understanding and incorporating the system’s behavior. Our work aligns with this perspective, placing significant emphasis on capturing the deformation nuances of digitally reconstructed fish. This approach represents a step forward in the quest for more comprehensive and behaviorally faithful digital twins.

### III. METHODOLOGY

The goal of this work is to develop a tool that zoologists can use to have access to the hydrodynamic maps in a water tank with a moving fish. The different tasks are therefore to segment the fish, using a video input (III-A), analyze the result of the



Fig. 1: Proposed pipeline for the automation of the design of our DT in LilyPad.

segmentation to extract the important features of the motion (III-B), and finally, to implement them in the CFD software (III-C), which is used in the global project described in the introduction. Fig 1 sums up our approach.

#### A. Segmentation with YOLOv8

This subsection aims to provide a concise overview of the implementation of the YOLOv8 algorithm for segmentation in our study. Rather than delving into an in-depth explanation of the inner workings of YOLOv8, we treat it as a black-box tool without conducting any network customization. Our focus is twofold: justifying the selection of YOLOv8 over other versions and offering insights into the training process.

The choice of YOLOv8 as our preferred algorithm is grounded in its demonstrated effectiveness and advancements over earlier versions. While we refrain from an exhaustive comparison, as it has already been done in others articles such as [7], YOLOv8’s superior performance in terms of accuracy, speed, and versatility influences our decision. Its robust object detection capabilities and compatibility with diverse applications align seamlessly with the multifaceted requirements of our study.

In this study, our primary goal is to demonstrate the feasibility of creating a dynamic digital twin using the YOLOv8 algorithm. The training approach is intentionally simplified, avoiding extensive data augmentation and adhering to default parameters. We use the *yolov8n-seg* model with a batch size of 16. This deliberate simplification showcases YOLOv8’s baseline capabilities for generating digital twins that emphasize behavioral nuances, underscoring its out-of-the-box effectiveness for our specific application. Further details on our dataset are discussed in the following paragraph.

To facilitate a robust comparison of the output generated by our pipeline, it is imperative to establish a ground truth. In pursuit of this, we opt to generate a simulation of a flexible body within the LilyPad CFD software. This simulated fish-like body serves as the input for our digital twin construction, as it has a similar behavior to the fish studied in our project. To train YOLO for the segmentation task, we employ a two-step process. First, we utilize various frames from LilyPad’s *FlexNACA* CFD simulations, providing diversity in conditions. Next, we create custom labels for training by employing OpenCV. This involves generating masks based on pixel color, effectively delineating the flexible fish within each frame. These custom labels not only provide the necessary ground truth for training YOLO but also enable the algorithm to learn the segmentation of the flexible body within different simulated environments. The following pictures (Fig.2) are taken

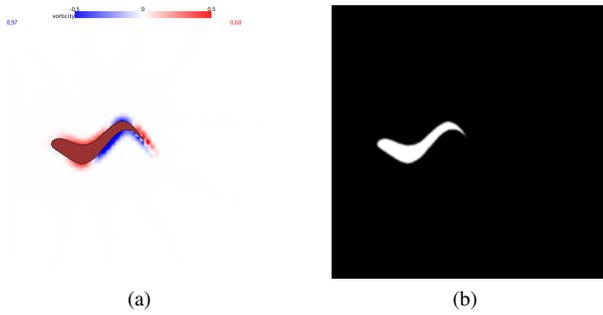


Fig. 2: Example of frame from the dataset and its corresponding mask for label

from the dataset and represent a frame and its segmentation mask before behind converted to yolo format to create a label.

### B. Motion processing

Upon obtaining YOLO’s predicted mask for each frame in the video, a subsequent series of Python-based steps is undertaken to ensure a smooth and continuous representation of the fish. This is crucial for compatibility with the LilyPad CFD software, which is sensitive to sharp angles and fast, noisy movements. This smooth representation is necessary to guarantee the convergence of the simulation.

YOLO generates a list of points corresponding to the edges of the flexible body shape within each frame. However, due to variations, the number of points may differ between frames. To address the variability in point numbers, a standardization process is initiated. This involves setting a predefined number of points, followed by rolling the list to position the smallest value as the first element. Duplicate points are then eliminated to ensure data consistency. Achieving a coherent representation of the fish shape involves parametric interpolation. Leveraging `scipy`’s `splprep` function, with default parameters except for the smoothing factor equal to half the number of coordinates and periodicity set to True, the rolled and de-duplicated point list is processed. Subsequently, the `splev` function is employed to perform parametric interpolation. This step is crucial for creating a smooth and continuous representation of the shape’s edges. The final output of the interpolation undergoes a normalization process to meet the specific format requirements of the CFD software. This step ensures seamless integration of the predicted mask into the simulation environment. Finally, unlike the spatial aspect of the prediction where interpolation facilitates the use of a straightforward finite differencing approach, thus directly computed in Digital Twin construction phase, the temporal consistency has yet to be addressed. Given the complexity introduced by temporal dynamics, a different differentiating method is required to ensure CFD convergence. Hence, the choice to employ the `SmoothedFiniteDifference` function from the `pySINDy` package [15], which provides a tailored solution for capturing temporal variations in noisy scenarios, by applying a Savitzky Golay smoother beforehand.

By meticulously executing these steps, the predicted mask from YOLO is transformed and refined into three csv files

for the x and the y coordinates and the temporal derivatives for every points in space and time, rendering it suitable for integration into the CFD software. This process not only ensures consistency across frames but also provides a dynamic and accurate representation of the flexible shape’s behavior in the digital twin environment.

### C. CFD parametrization

In this section, we outline the Computational Fluid Dynamics (CFD) parametrization employed for the construction of the digital twin, based on the architecture of the `FFlexNACA` example provided in the LilyPad library. Lilypad is a 2D low-fidelity CFD software that uses the Boundary Data Immersion Method (BDIM) to solve the Navier-Stokes equation [16], independently from the shape that is being simulated as it is not sensitive to the grid [16]. It is preferred to some other CFD solutions for its fast computing and simplicity, though it might not be the most precise solver.

A custom class named `CSV2DigitalTwin` is devised to adapt the `FlexNACA` framework to accommodate the unique requirements of our digital twin. Key distinctions lie in the utilization of three CSV files obtained from the previous phase, enabling a more comprehensive representation. This class takes the previously mentioned files as input. Initialization involves loading these files, establishing initial parameters, and creating instances for further computation. A crucial step is the computation of spatial derivatives using the loaded CSV files. This is necessary for the integration of LilyPad’s inbuilt functions and essential for accurately simulating the dynamic behavior of the deformable fish shape and its impact on the fluid. The digital twin is constructed by placing points corresponding to the first time step onto the canvas. These points serve as the initial configuration of the digital twin. Subsequently, as time progresses, the positions of these points are updated based on the information extracted from the CSV files. This iterative process generates a deformable fish, mirroring the motion observed in the original `FlexNACA` simulation. The enhanced capabilities of the `CSV2DigitalTwin` class enable the redefinition of inbuilt LilyPad functions tailored for fluid mechanics simulations. These include:

- **WallNormal:** Providing the adjusted normal vector to the base shape, considering the local deformations introduced by the digital twin.
- **Velocity:** Calculating the velocity of the digital twin, incorporating the effects of both the base shape and the deformations introduced over time.
- **Distance:** Determining the distance between points on the digital twin and the rest of the canvas, accounting for spatial variations induced by the system’s dynamics.

The digital twin evolves dynamically over time, capturing the intricate interplay between the deformable fish shape and the surrounding fluid. This dynamic evolution is achieved through continuous updates to the coordinates of the digital twin based on the information extracted from the CSV files, ensuring a faithful representation of the original simulation’s fluid-structure interactions. To complete the simulation setup,

it is essential to define the remaining parameters, including the canvas size, viscosity, and time step. In order to maintain consistency and facilitate a direct comparison with the output of the proposed pipeline, we opt to retain the parameters from the original *FlexNACA* simulation. These parameters, initially set as the default values provided by LilyPad, ensure a standardized environment for the comparison, aligning with the conditions of the reference simulation.

This CFD parametrization framework provides a robust foundation for the digital twin, allowing it to mirror the behavior of the flexible fish shape in a fluid flow. The subsequent sections will delve into the performance metrics, validation against ground truth, and insights gained through the application of this parametrization methodology.

#### IV. RESULTS

##### A. Training results

The initial YOLO training utilizes a dataset comprising 500 images, divided into a training set and a test set with an 80/20 ratio. Examination of the training and test losses reveals a positive outcome, indicating that the model is not overfitting, see Fig.3. This initial observation marks a promising starting point for further refinement. However, upon a deeper analysis of additional metrics (mean average precisions mAP50 and mAP50-95), it becomes apparent that the model could potentially benefit from more extensive training data. Metrics, while satisfactory, show room for improvement, as they deviate notably from the ideal value of 1 (Fig.3). Recognizing the potential for enhanced learning, a decision is made to retrain the model with a larger dataset, specifically 30,000 frames. It is worth remembering that while this training may not adhere to the most rigorous standards, it serves the purpose of exploring the model's potential (III-A).

The outcomes of the retraining validates the intuition. The losses remains favorable (Fig.4), indicating effective training without overfitting. Notably, mAP50 and mAP50-95 shows marked improvement, approaching values close to 1. This shift in metrics suggests that the increased dataset indeed contributes to a better understanding of the object detection task. Interestingly, the metrics reaches a plateau early in the training while the validation loss doesn't decrease as fast as the training one. This observation prompts consideration for future studies to diversify the dataset further, introducing a broader range of scenarios and complexities to enhance the model's robustness and the digital twin's performance, as it will be highlighted in the subsequent sections.

In summary, the YOLO training results demonstrate a positive trend (Fig. 5). The transition from the initial dataset to a larger one led to improved metrics, reaffirming the model's capability for object detection. The pursuit of a more diverse dataset remains a logical next step, aiming to address potential limitations and further elevate the model's performance.

##### B. CFD results under static flow

In order to compare the original simulation with the digital twin, a static flow scenario is established (Fig.6). This con-

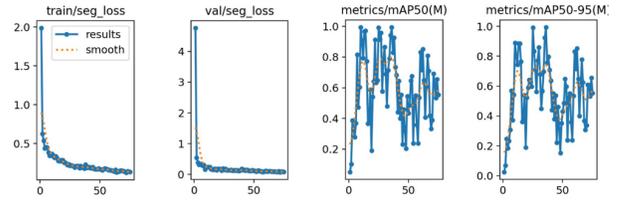


Fig. 3: Training metrics provided by YOLOv8 for a 500 frames dataset, with a 80/20 split between train and test sets.

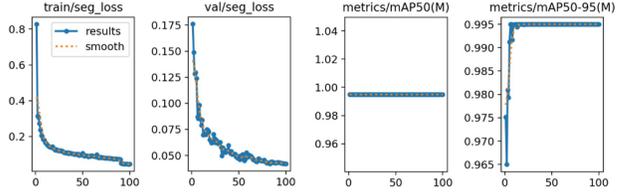


Fig. 4: Training metrics provided by YOLOv8 for a 30 000 frames dataset, with a 80/20 split between train and test sets.

figuration serves as the baseline for the original simulation. Notably, the mean pressure field of the digital twin demonstrates intriguing similarities to the original simulation, albeit with reduced intensity. A notable discrepancy emerges in the nose region of the fish, warranting further investigation.

To delve into this observed difference, various simulations of the digital twin are conducted with combinations of customizations, and for each of them, metrics such as the mean pressure, different norms or variance are computed (Tab.I). For instance, individual elements extracted from the YOLO output, such as the spatial derivative, are implemented while retaining the temporal derivative and distance function from the original simulation. This sensitivity analysis aims to pinpoint the most impactful customizations influencing the discrepancies observed. Results from the sensitivity checks shed light on the significance of the temporal derivative as the primary factor influencing the observed differences. Its impact is nevertheless reduced when coupled with all the other customization. This insight provides valuable guidance on areas for potential improvement in the digital twin simulation, while still keeping in mind the improvements that could come out of a proper training.

In summary, the comparison between the original simulation and the digital twin in a static flow scenario reveals a generally comparable mean pressure field. However, specific differences



Fig. 5: Original shape (a) vs the reconstructed one using YOLOv8 output (b)

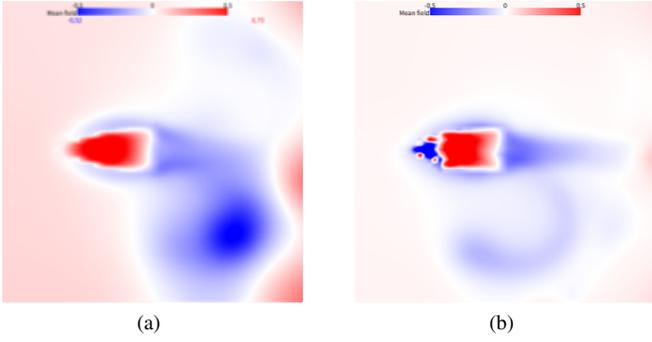


Fig. 6: Mean pressure field measured in the ground truth simulation (a) and in the digital twin simulation (b), both in a static flow.

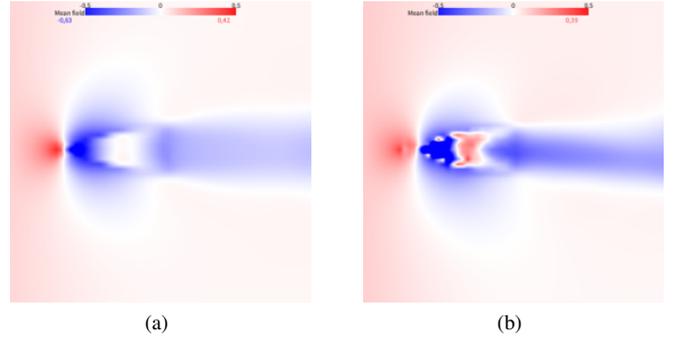


Fig. 7: Mean pressure field measured in the ground truth simulation (a) and in the digital twin simulation (b), both in a forced flow.

in the nose region prompts a deeper exploration through sensitivity analyses. The findings underscore the importance of the temporal derivative as a key element influencing the simulation outcomes, guiding future refinement efforts for enhanced fidelity in the digital twin representation.

TABLE I: Pressure field metrics for all combinations of customization using spatial (dhdxd), temporal (hdot) derivation and distance (dist) function, in a static flow.

	mean	$l_1$ norm	$l_2$ norm	$l_\infty$ norm
<b>Custom shape</b>	96.3	97.9	98.6	108
<b>dhdxd</b>	71.5	99.0	98.2	108
<b>hdot</b>	93.9	111	167	568
<b>dist</b>	51.4	94.0	93.5	69.2
<b>dhdxd &amp; hdot</b>	91.9	111	167	568
<b>dhdxd &amp; dist</b>	45.7	93.2	92.6	68.7
<b>hdot &amp; dist</b>	56.4	94.7	122	311
<b>all</b>	50.2	95.0	122	312
<b>all (forced flow)</b>	135	145	332	8450

All values are percentages of original simulation metrics.

### C. CFD results under forced flow

Recognizing the limitations in the initial scenarios, efforts are directed towards creating more realistic conditions for the comparison between the original simulation and the digital twin. The key adjustment involves increasing the size of the water tank to more accurately simulate real-world conditions. By virtually enlarging the tank by using a forced flow, the goal is to allow perturbations generated by the fish’s motion to unfold in a more natural manner, minimizing interactions with the tank boundaries and enhancing the overall realism of the simulation. The results from these refined scenarios, while still displaying qualitative differences, exhibit improvements compared to the initial experiments. The adjustments in flow speed contributes to a more realistic representation of the interaction between the fish and its environment. Although differences persist, they are observed in a less pronounced manner on the mean pressure field (Fig.7), offering a clearer insight into the nuanced dynamics between the original simulation and the digital twin. The metrics, on the other hand, present an

alternative perspective. While variations are observed in the mean and  $l_1$  norm between the two setups, the most notable shifts are evident in the  $l_2$  and  $l_\infty$  norms. In the case of an open tank, these metrics experience a significant surge, reaching differences of 332% and 8450% between the original and digital twin, in contrast to 122% and 312% for the closed tank (Tab.I). This suggests the possibility of pressure outliers in specific areas of the digital twin simulation, which once again could be the nose area, consequently resulting in more substantial discrepancies in metrics that emphasize variations, such as the  $l_2$  and  $l_\infty$  norms.

### D. CFD results for a flow with $U_\infty = 1$ in a wide tank

Our exploration into an open tank scenario prompts an intriguing investigation into the influence of a larger observation window, four times the size of the original. Utilizing the same parameters as before, but with an expanded spatial scope, we shift our focus to examining a specific frame of vorticity from the simulation rather than the mean pressure map. The results reveals a distinct vorticity pattern in the digital twin that deviates notably from the original simulation (Fig.8), particularly as we move further downstream the fish. While this experiment do not unveil new differences, it provides valuable insights into the cumulative impact of the earlier observed small discrepancies, particularly when applied to a dynamic and chaotic environment. In essence, this experiment highlights that, despite qualitative similarities between the digital twin and the original simulation, the reconstructed system’s simulation will inherently tend to diverge over time. However, it is essential to approach these results with a degree of caution. The comparison involves the digital twin simulation of a fully controlled system against its ground truth. In a real-world scenario, such as digitalizing a living fish, the absence of a ground truth complicates the evaluation. In addition, no living fish has such a regular and symmetrical alternating movement creating such wake. Also a realistic turbulence level of the water causes more dissipation, which is not considered here. The alternative approach would be manual design of the digital twin, a process inherently prone to introducing subtle differences from reality due to the inherent

challenge of capturing the entirety of a system’s dynamics. In essence, both approaches result in a similar situation, but the manual design method demands significantly more time for simulation construction, thus emphasizing the efficiency and practicality of the proposed YOLO-based method, without underestimating the remaining necessary improvements.

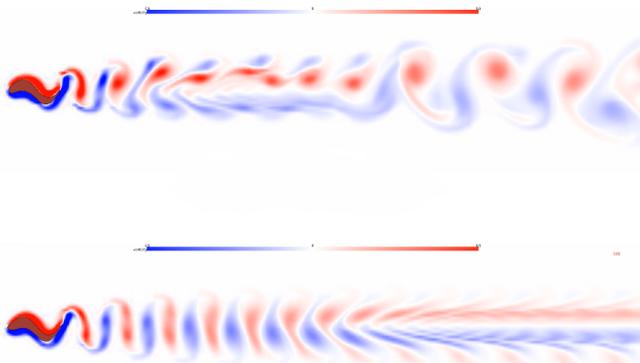


Fig. 8: Vorticity maps for the digital twin (top) and the original simulation (bottom) in a wider tank after 28 tailbeats.

## V. CONCLUSION

In this study, we embark on a comprehensive exploration of enhancing digital twin simulations through the integration of YOLO-based object detection, focusing on the motion dynamics of a flexible body in fluid flow scenarios. By employing YOLO as a segmentation model, we successfully reconstruct the shape and motion of the fish, offering a dynamic representation for digital twin simulations.

This approach presents a notable improvement over previous methods, as it leverages state-of-the-art computer vision techniques to capture the intricate details of the system’s behavior and fast deformation. The segmentation model, trained on a relatively small dataset, exhibits promising results in object detection, enabling a faithful recreation of the fish’s motion. Moreover, the incorporation of sensitivity analyses and realistic scenarios allowed us to identify key factors influencing the simulation outcomes. While the digital twin demonstrates qualitative proximity to the original simulation, the study shed light on nuanced differences, particularly in dynamic and chaotic environments. Notably, the impact of temporal derivatives emerges as a critical aspect influencing simulation fidelity. To further refine this digital twin design, future efforts should prioritize the incorporation of a more diverse dataset, encompassing a range of scenarios and shapes of body to enhance the model’s robustness. Additionally, addressing the observed differences in dynamic environments requires to investigate more powerful methods to extract the temporal derivatives from the raw data. Tools such as *SINDy* from the *pySINDy* package are being considered. Looking ahead, one exciting avenue for exploration is the application

of these methods to real fish videos. This next step promises to bridge the gap between simulations and real-world scenarios, offering opportunities for validating and refining the digital twin approach in actual, uncontrolled environments.

In conclusion, this study represents a significant advancement in digital twin simulations, showcasing the potential of YOLO-based object detection in capturing the complexities of dynamic systems. By understanding the uncovered limitations and nuances, we lay the foundation for continued refinement, aiming for a digital twin solution that not only qualitatively mirrors the original simulation but also adapts to the challenges of dynamic and evolving scenarios, including real-world fish behaviors.

## REFERENCES

- [1] ultralytics github
- [2] Weihs, D. (1975). Some hydrodynamical aspects of fish schooling. In *Swimming and flying in nature* (pp. 703-718). Springer, Boston, MA.
- [3] Drucker, E. G., & Lauder, G. V. (2000). A hydrodynamic analysis of fish swimming speed: wake structure and locomotor force in slow and fast labriform swimmers. *Journal of Experimental Biology*, 203, 2379-2393.
- [4] Videler, J. J., Stambhuis, E. J., Müller U. K., & Van Duren, L. A. (2002). The scaling and structure of aquatic animal wakes. *Integrative and comparative biology*, 42(5), 988-996.
- [5] Coombs, S.m Janssen, J., & Webb, J. F. (1988). Diversity of lateral line systems: evolutionary and functional considerations. In *Sensory biology of aquatic animals* (pp. 553-593). Springer, New York, NY.
- [6] Liao, J. C., Beal, D. N., Lauder, G. V., & Triantafyllou, M. S. (2003). Fish exploiting vortices decrease muscle activity. *Science*, 302, 1566-1569. <https://github.com/ultralytics/ultralytics>
- [7] Hussain, Muhammad. *YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection*. *Machines* 11, no. 7 (June 23, 2023): 677.
- [8] Lassoued, Khaoula, Tonino Sophy, Julien Jouanguy, and Luis Le-Moyne. *Fluid Flow Simulation over Complex Shape Objects Using Image Processing to Achieve Mesh Generation*, n.d.
- [9] Doroshchenko, I.A., I.A. Znamenskaya, N.N. Sysoev, and A.E. Lutsky. *High-Speed Flow Structures Detection and Tracking in Multiple Shadow Images with Matching to CFD Using Convolutional Neural Networks*. *Scientific Visualization* 14, no. 4 (2022).
- [10] Zhou, Xiaoping, Kaiyue Sun, Jia Wang, Jichao Zhao, Chiyuan Feng, Yalong Yang, and Wei Zhou. *Computer Vision Enabled Building Digital Twin Using Building Information Model*. *IEEE Transactions on Industrial Informatics* 19, no. 3 (March 2023): 2684-92.
- [11] Sundby, Tiril, Julia Maria Graham, Adil Rasheed, Mandar Tabib, and Omer San. *Geometric Change Detection in Digital Twins*. *Digital* 1, no. 2 (April 15, 2021): 111-29.
- [12] Du, Baoxia, Hongyang Du, Haifeng Liu, Dusit Niyato, Peng Xin, Jun Yu, Mingyang Qi, and You Tang. *YOLO-Based Semantic Communication with Generative AI-Aided Resource Allocation for Digital Twins Construction*. arXiv, June 25, 2023.
- [13] Zhou, Xiaokang, Xuesong Xu, Wei Liang, Zhi Zeng, Shohei Shimizu, Laurence T. Yang, and Qun Jin. *Intelligent Small Object Detection for Digital Twin in Smart Manufacturing With Industrial Cyber-Physical Systems*. *IEEE Transactions on Industrial Informatics* 18, no. 2 (February 2022): 1377-86.
- [14] Hu, Weifei, Tongzhou Zhang, Xiaoyu Deng, Zhenyu Liu, and Jianrong Tan. *Digital Twin: A State-of-the-Art Review of Its Enabling Technologies, Applications and Challenges*. *Journal of Intelligent Manufacturing and Special Equipment* 2, no. 1 (August 10, 2021): 1-34.
- [15] Kaptanoglu, Alan, Brian De Silva, Urban Fasel, Kadierdan Kaheman, Andy Goldschmidt, Jared Callahan, Charles Delahunt, et al. *PySINDy: A Comprehensive Python Package for Robust Sparse System Identification*. *Journal of Open Source Software* 7, no. 69 (January 29, 2022): 3994.
- [16] Maertens, Audrey P., and Gabriel D. Weymouth. *Accurate Cartesian-Grid Simulations of near-Body Flows at Intermediate Reynolds Numbers*. *Computer Methods in Applied Mechanics and Engineering* 283 (January 2015): 106-29.