Depth Estimation of Hydro-Foiling Craft Using Computer Vision

Andre R. Cox, Cara Mikoliunas, Rayman Klar, and Lina Naleto

University of British Columbia

ENGR 400 Applied Machine Vision for Engineers

Dr. Zheng Liu

March 16, 2026

Author Note

Detailed author responsibilities are provided in Appendix A, Section *Author Contributions*.

Abstract

Ride height is a critical control variable for high-performance hydrofoiling craft (e.g., foiling yachts and electric hydrofoils). Current systems typically estimate ride height using LiDAR, ultrasonic ranging, or a combination of both. However, each sensing method has practical drawbacks in hydrofoiling environments. Ultrasonic sensors require careful placement to avoid returns from the hull and structure. These sensors are sensitive to spray, aerated water, and wind; at high speeds, they may produce intermittent or unreliable measurements (Rak, Hocevar, Kolbl Repinc, Novak, & Bizjan, 2023). LiDAR can provide faster updates and higher resolution, but performance can degrade due to spray and surface-reflection effects, and measurements may be biased when the sensor samples only a small patch of the water surface (Rak et al., 2023). To address these limitations, we propose a vision-based ride-height estimation method that uses a high-speed camera to capture images of a hydrofoil mast. A computer vision algorithm segments the mast and detects the air–water interface along it to estimate ride height. This approach aims to provide a low-cost, non-contact estimate that remains reliable under spray and high-speed operating conditions.

Table of Contents

Introduction

A hydrofoil craft is a boat with wing structures mounted below the hull. These structures are responsible for creating lift as the boat gains speed and consequently raising the hull above the surface. Since the hull stays above the water's surface, the contact area with the water is reduced, which makes the hydrofoil more fuel-efficient compared to other boats (Boatsetter Team, 2024).

An essential part of the hydrofoil's control system is real-time ride-height measurement. The ride height influences the hydrodynamic forces generated by the foils, which affect the boat's stability, efficiency, and safety. The control system depends on continuous height feedback to adjust control surfaces or foil angles. If the measurement is delayed, noisy, or inaccurate, the control system may respond improperly, potentially leading to increased drag, loss of efficiency, or unstable flight conditions.

A few different sensing technologies are currently used to measure the distance to a water surface in marine applications. One option is to use an ultrasonic sensor. These sensors emit acoustic waves toward the surface and estimate distance from the returning signal. Ultrasonic sensors provide a moderate frequency response of 20 Hz for the ToughSonic® 100 14' Range, 1" NPT (Senix Corporation, 2026b), which is used for Emirates New Zealand – 2017 winner in America's Cup racing (Senix Corporation, 2026a). However, they are sensitive to air bubbles, particles, and turbulence in the flow, which can compromise measurement reliability. In highly aerated or turbulent flows, ultrasonic sensors may capture only average surface characteristics rather than accurately reproducing the full surface dynamics (Rak et al., 2023).

Another technology used for measuring water surface elevation is LiDAR (Light Detection and Ranging). LiDAR systems determine distance by emitting laser pulses toward the surface and measuring the time it takes for the reflected light to return. Modern LiDAR devices can perform measurements at very high frequencies and provide detailed information about the water surface topography. However, LiDAR measurements

can also be affected by reflections from bubbles, droplets, or particles, which introduce noise in the estimated distance to the water surface (Rak et al., 2023).

Vision-based sensing offers another way to measure the position of the water surface. In engineering measurements, cameras are often chosen because they can capture detailed images at a relatively low cost. Unlike ultrasonic or LiDAR sensors, which emit signals to gather data, cameras instead rely solely on captured images. Building on this, computer vision techniques process these images to detect useful features, such as edges and boundaries. Notably, the boundary between air and water can be identified in images, enabling estimation of the water surface level without direct contact.

This report explores a vision-based approach to estimating ride height for hydrofoil boats. The proposed method uses a high-speed camera to capture images of the hydrofoil mast and applies computer vision techniques to detect the air-water interface along the mast. By identifying this boundary in each frame, the system's ride height can be estimated. The next sections describe this approach.

Project Description

This project involves generating a training dataset, as none currently exists for this specific machine-vision problem. The dataset is generated by simulating the target scenario in Blender, utilizing BlenderProc to procedurally create thousands of photorealistic synthetic images. This dataset is subsequently used with two neural network approaches: (1) a custom model developed using TensorFlow and (2) a modified ResNet-18 convolutional neural network adapted for regression by replacing the final classification layer with a single continuous ride-height output from RGB frames.

To support the neural network approaches, a well-established architecture was adopted. ResNet-18 is a convolutional neural network (CNN) that employs deep residual learning, where a stack of layers learns a correction rather than an entirely new transformation (He, Zhang, Ren, & Sun, 2016). The network consists of residual blocks, with each block output formed by adding the block input, x, to the learned residual

function, $F(x)$, resulting in $y = F(x) + x$ (He et al., 2016). This skip connection improves gradient flow and training robustness (He et al., 2016). ResNet-18 features an 18-layer architecture and is widely used as a compact backbone for image-processing applications (He et al., 2016).

ResNet-18 was selected over segmentation models such as YOLOv8 because those models require bounding boxes or segmentation labels, whereas the dataset contains only RGB images and scalar ride height labels (Hussain, 2023). The possibility of generating pre-segmented data with BlenderProc was considered; however, this approach was not pursued, as object detection was not the focus of the study. Consequently, ride height estimation was treated as a regression problem rather than a detection task, making a ResNet-based regression baseline more appropriate for the available label format.

Methodology

BlenderProc Data Generation Pipeline

As there is no existing training data for this project, a dataset was to be synthetically generated. This was done using BlenderProc, a procedural generation pipeline for Blender3D (Denninger et al., 2023). The advantage is that it allows a large amount of varied data to be generated across novel environments and lighting conditions. The location of the camera was placed as if it were attached to the body of the Hydrofoil, pointing down towards the water at an approximately 56° angle.

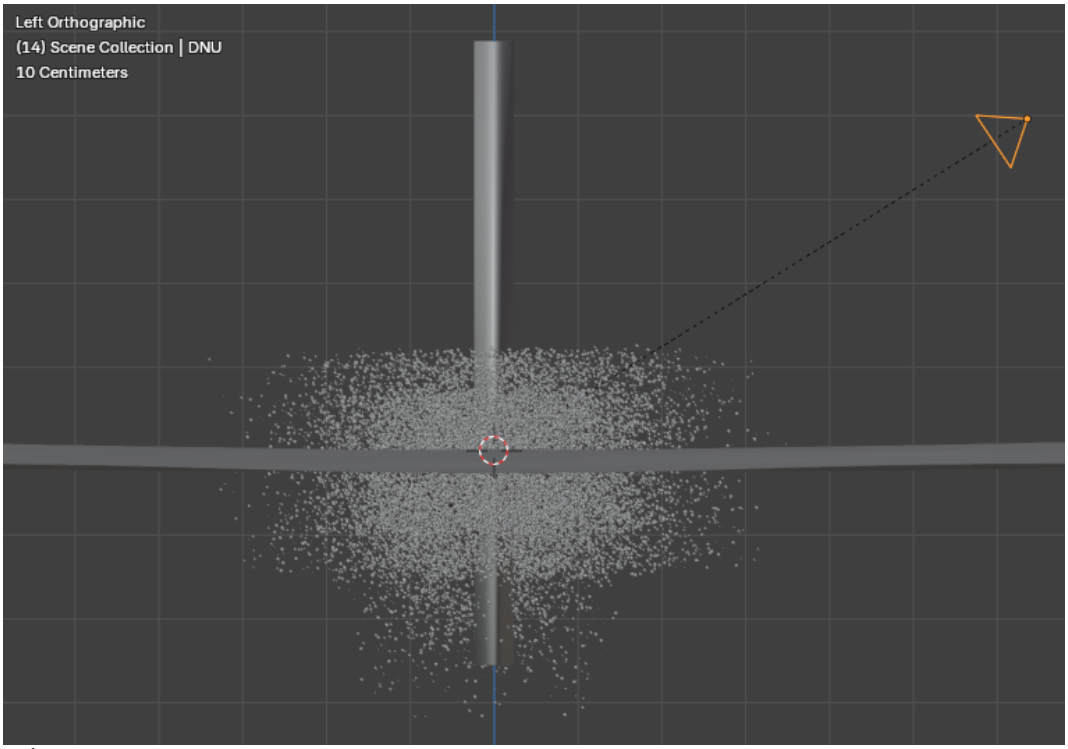

Figure 1

Layout of the simulated scene. Observe the particle system centered around the 3D cursor to simulate water spray.

A geometry node-based particle system was used, connected to a dynamic paint system to simulate a wake and water spray. This was done to achieve better photorealism while reducing computational cost (Alzu'bi, Al-Ayyoub, Jararweh, & Gupta, 2022). The camera was also set up to simulate a GoPro Hero 5. This choice was made, as it is a relatively inexpensive camera with high frame rates (240 fps at 720p). It is also rugged and water-resistant, which is important in a marine environment. Motion blur was simulated too; however, this was adjusted artistically, and no calculations were done to make this accurate.

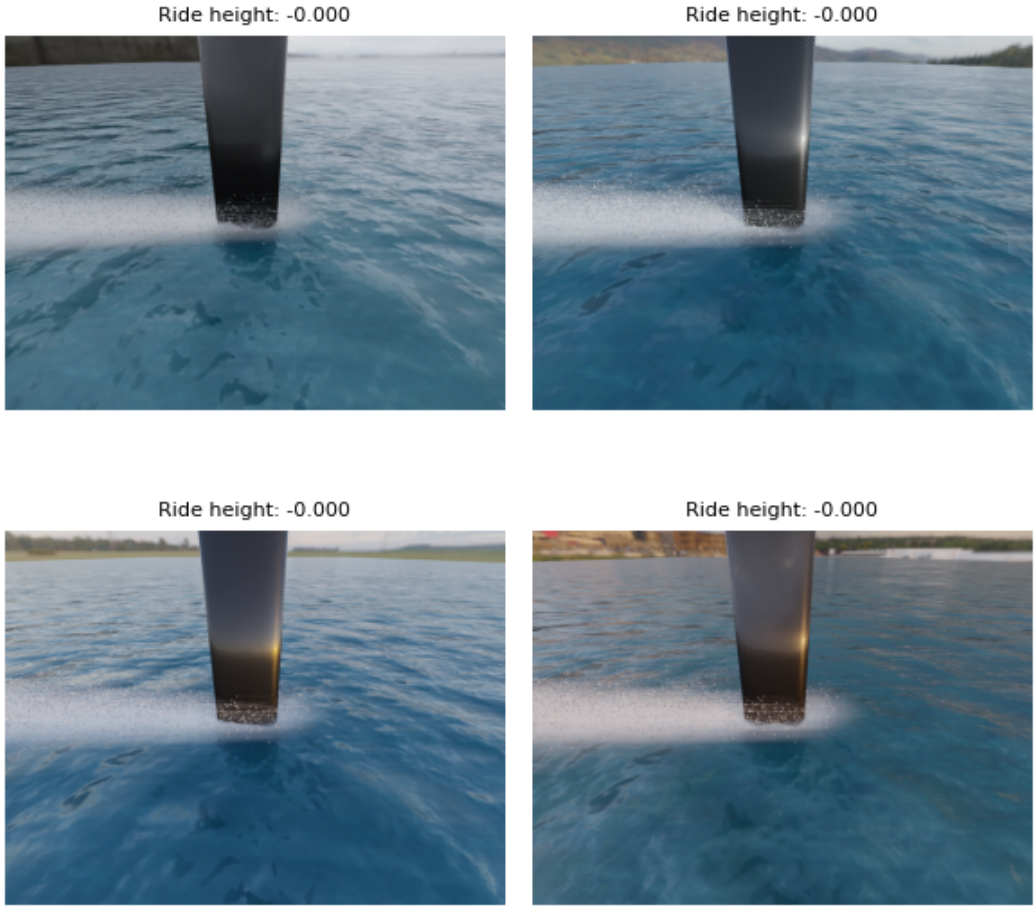

Figure 2

Sample synthetic training frame generated using BlenderProc.

The total time required to generate the dataset was 7:40:23 for 3200 images at a resolution of $960x720$, corresponding to roughly 8.5 seconds per image on an NVIDIA RTX 5070 Ti. Rendering was performed with the Cycles engine to capture realistic bounce lighting and reflections, using 64 ray-tracing samples per frame. Although this sample count is lower than typical photorealistic settings, OpenImageDenoise was applied to reduce graininess in the final outputs. After rendering, BlenderProc packaged each sample into an individual .hdf5 file containing the image and associated metadata fields listed in Table 1 below.

Table 1

Metadata Stored Per Synthetic `.hdf5` Sample

Field	Type / Shape	Description
`colors`	Dataset `{720, 960, 3}`	RGB image tensor stored per sample.
`hdri_rotation`	Dataset `{SCALAR}`	HDRI rotation value used for the rendered frame.
`hdri_source`	Dataset `{SCALAR}`	HDRI source file name for the rendered frame.
`ride_height`	Dataset `{SCALAR}`	Ground-truth ride-height label normalized between 0 and 1.
`velocity`	Dataset `{SCALAR}`	Simulated craft velocity label (unused for this report).

Data Preprocessing and Analysis

After image generation, 3200 .hdf5 files were produced, totalling approximately 3.0 GB. To reduce storage and improve I/O efficiency, additional compression was applied. Although HDF5 supports lossless compression and had already reduced the dataset size from about 6 GB to 3 GB, a lossy approach was selected for further reduction. WebP was chosen because it provides a high compression ratio while maintaining acceptable visual quality. A processing script compressed the images and consolidated them into a single HDF5 file for faster loading, since handling thousands of small files is computationally expensive for the CPU.

Table 2

Final Storage Comparison After WebP Compression

Metric	Value
Original folder size	3.23GB
New HDFS (WebP) size	0.12GB
Net space saved	96.21%

ResNet-18 Regression Baseline

As a baseline model, we used a ResNet-18 backbone and replaced its final classification head with a single-output regression layer. The model input is an RGB frame generated by the BlenderProc pipeline, and the target output is the normalized ride-height value associated with that frame. Each sample is stored as an `.hdf5` file containing an RGB image (`colors`, `720x960x3`, `uint8`) and scalar labels. In this dataset, ride height is normalized to the range [0, 1]. Before inference with ResNet-18, images are resized to `224x224` and normalized using standard ImageNet channel statistics to ensure compatibility with pretrained ResNet weights (PyTorch Contributors, 2026). This baseline provides a strong reference point because residual architectures are well established for visual feature extraction and can learn robust geometric cues from synthetic data.

For training, the dataset is split into training and validation subsets with an 80/20 split to monitor generalization during optimization. Training minimizes mean squared error (MSE) loss using the Adam optimizer for 20 epochs, and performance is reported using both mean absolute error (MAE) and root mean squared error (RMSE). MAE is used as the primary evaluation metric because it is directly interpretable as a measure of ride-height prediction error. During training, we track both training loss and validation loss per epoch and retain the model checkpoint with the best validation performance.

The baseline serves two purposes in this project. First, it provides a reproducible benchmark against which the custom model can be compared. Second, it helps identify

whether prediction limitations come from model architecture or from data-generation assumptions such as lighting variation, camera placement, and wake appearance.

Custom CNN in Keras

In parallel with the ResNet-18 baseline, we developed a custom convolutional neural network in TensorFlow/Keras tailored to scalar ride-height regression. The custom network is designed to remain compact while capturing features relevant to hydrofoil depth estimation, including mast boundaries, waterline position, and spray patterns near the hull.

The model is trained on the same synthetic dataset and evaluated with the same validation protocol as the baseline to support a fair comparison. We use Keras training logs to record loss and MAE over time, then summarize convergence behaviour and final validation performance. This consistency allows differences in results to be attributed primarily to architectural choices rather than training procedure.

After developing a simple CNN, the MAE was deemed to be unacceptably high. To correct this, the model was updated to use residual layers instead. This significantly improved the performance of the model. However, it did increase complexity significantly.

The custom CNN is included to assess whether a lighter, task-specific architecture can match or exceed the performance of a pretrained-style residual backbone while reducing computational cost. If successful, this approach may be advantageous for real-time deployment on embedded marine systems with limited onboard compute.

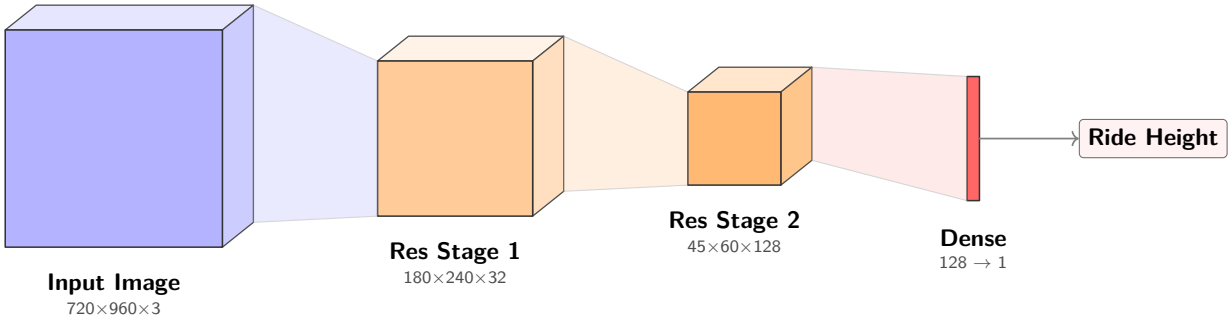

Figure 3

3D Architectural Pipeline for Ride-Height Regression

Note. The model processes high-resolution input (blue) through successive residual stages (orange), increasing feature depth while reducing spatial dimensions, culminating in a 1D regression output.

Data Avalability

The Blenproc synthetic data script, custom CNN and modified ResNet-18 architectures were implemented in Python 3.12. The complete source code, including the BlenderProc generation scripts, is available on GitHub Cox (2026).

Findings and Results

After both models had been trained, we used matplotlib to visualize the validation data in an understandable way. We also tested using novel data, which used unique HDRIs to simulate new environments and lighting conditions.

ResNet-18 Baseline

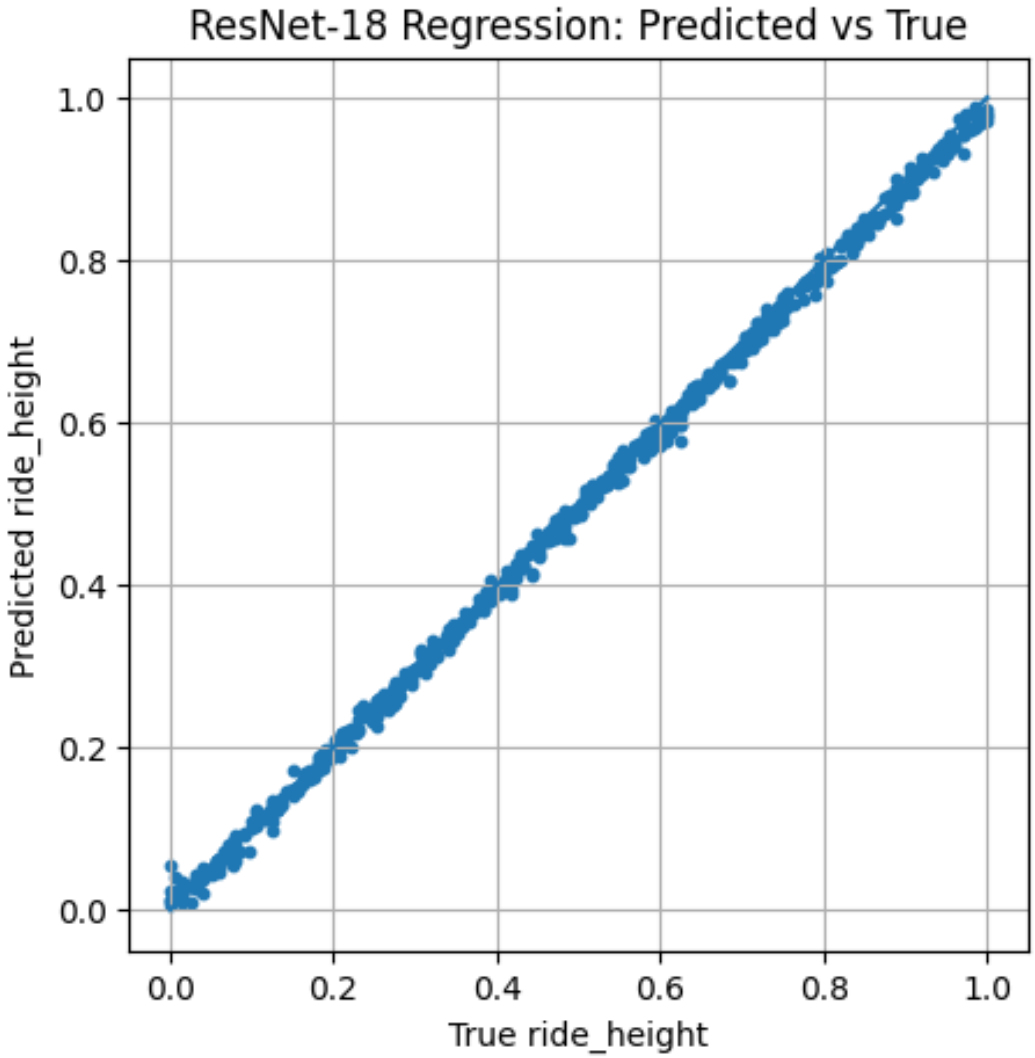

Figure 4

Correlation between true and predicted normalized ride-height values for the ResNet-18 baseline using data with the same backgrounds.

Figure 4 shows the initial predicted versus actual ride height for the ResNet-18 baseline trained on the WebP-encoded synthetic dataset (HDF5) generated by a simulation environment. In this simulation, the scenes were automatically rendered with consistent mast and water configurations, resulting in tightly clustered points along the $y = x$ line. This indicates that the model learned a strong mapping from RGB images to the scalar ride height label across the full normalized range (0–1). This behaviour is consistent with low regression error and suggests that the synthetic dataset contains a clear visual cue, specifically the mast's vertical position relative to the water. Although there is slight dispersion at the lower end of the plot, the overall trend is very strong, indicating that the model is learning the physical attributes needed to estimate height.

Building on these findings, it is important to note that this initial result was obtained using a dataset generated from simulated environments under relatively consistent visual conditions (similar scene configuration and ocean environment). Therefore, the observed performance mainly demonstrates that the regression model and its architecture can learn ride height when scene variation is limited. Further experiments should introduce greater diversity in wake appearance, camera pose, and potentially introduce nighttime lighting to evaluate model robustness and reduce the risk of overfitting to a single scenario.

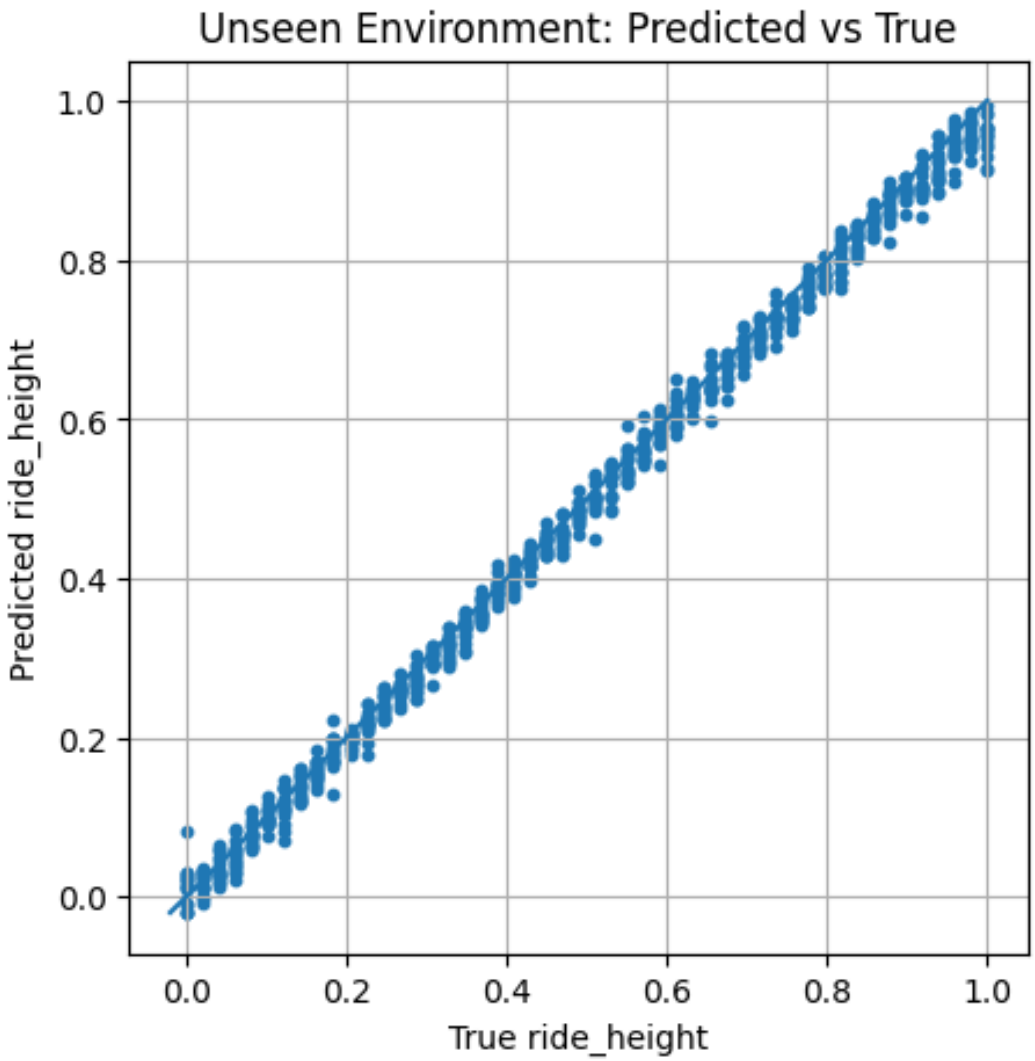

Figure 5

Correlation between true and predicted normalized ride-height values for the ResNet-18 baseline using data with novel background.

Figure 5 shows ResNet-18 performance after being trained on a dataset that includes novel background scenarios and a wider range of parameters. The plot still shows a tight spread around the $y = x$ relationship, similar to Figure 4. However, compared with the same-background case (Figure 4), the spread is wider, as expected. This wider spread means that the model is more sensitive to environmental changes than to ride-height variation alone. It also indicates that some learned visual features are partially tied to

scene appearance, such as lighting and background context, rather than only to the mast and water interface. Despite this, the model holds a strong positive trend between true and predicted values, demonstrating that it can preserve ride-height information under more challenging conditions.

Analyzing both cases together, Figures 4 and 5 show that the ResNet-18 baseline performs very strongly when training and validation conditions are kept visually similar; however, generalization becomes slightly more difficult when the background is varied. This comparison is important because a real-world hydrofoil sensing system would operate in varying outdoor conditions rather than in a single controlled environment. Therefore, the novel-background case provides a more realistic outcome of how the model would behave in the field.

Custom CNN

Like the ResNet-18 baseline, the custom CNN was trained and validated on the same dataset. Figure 6 shows the performance of the custom CNN when trained and validated on data with the same background conditions. The output exhibits a generally linear trend along the ideal correlation line. This shows that the custom architecture can learn key visual relationships between the mast and normalized ride height. These results confirm that a lighter, task-specific network can capture the main regression features required for training and validation. However, the spread is significantly worse than our ResNet-18 baseline, showing much worse performance at the upper and lower end of the ride-height spectrum.

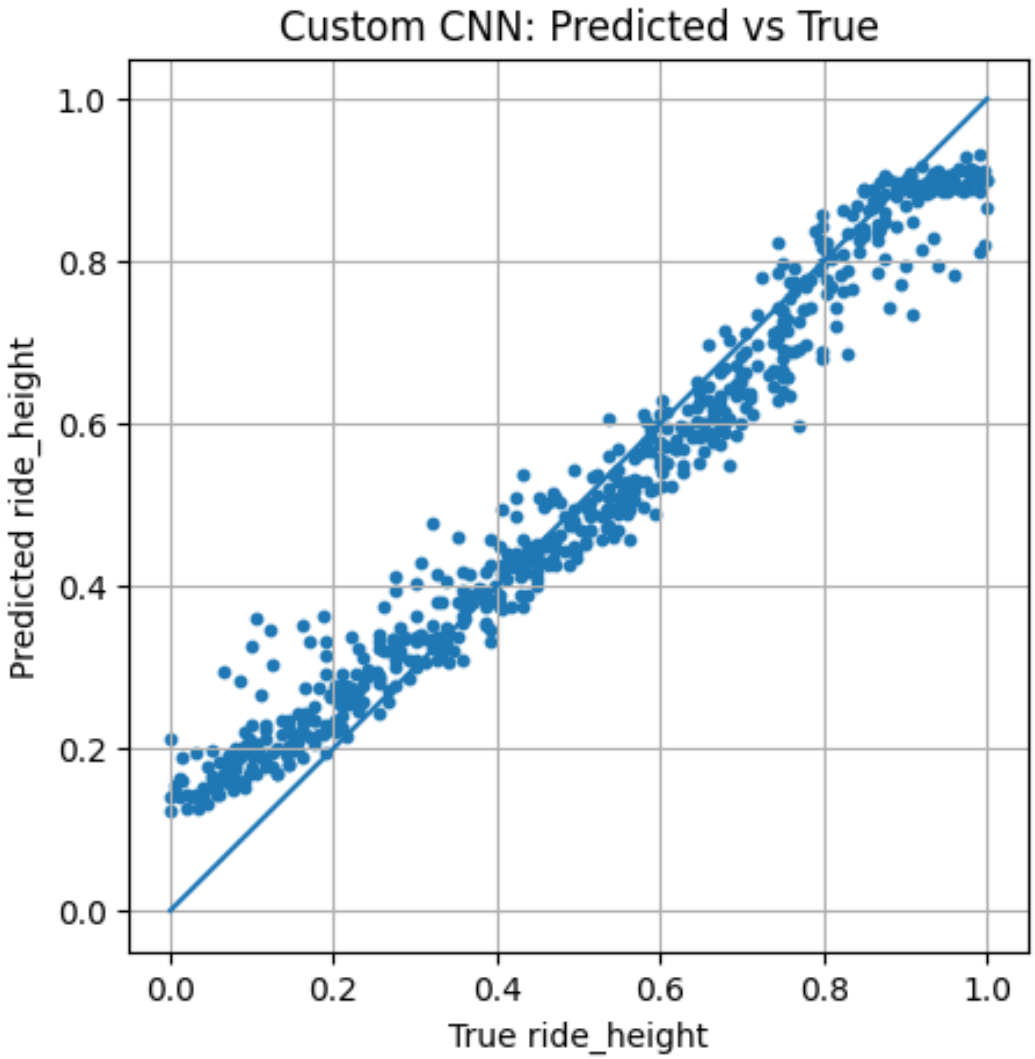

Figure 6

Correlation between true and predicted normalized ride-height values for the custom CNN using data with the same background.

Figure 7 shows the custom CNN evaluation under novel background conditions. The overall shape is generally similar to that of Figure 6, where the backgrounds were consistent; however, as observed with the ResNet-18 baseline, this plot exhibits a wider spread. This indicates that the model's regression accuracy is influenced by scene diversification, and that background variation remains a major source of generalization error. However, the overall upward trend in the plot shows that the custom CNN is still

extracting useful information and physical cues from the mast and surrounding waterline. Overall, this result from Figure 7 shows that the custom CNN can learn task-relevant features, but is less robust when the synthetic training dataset is diversified.

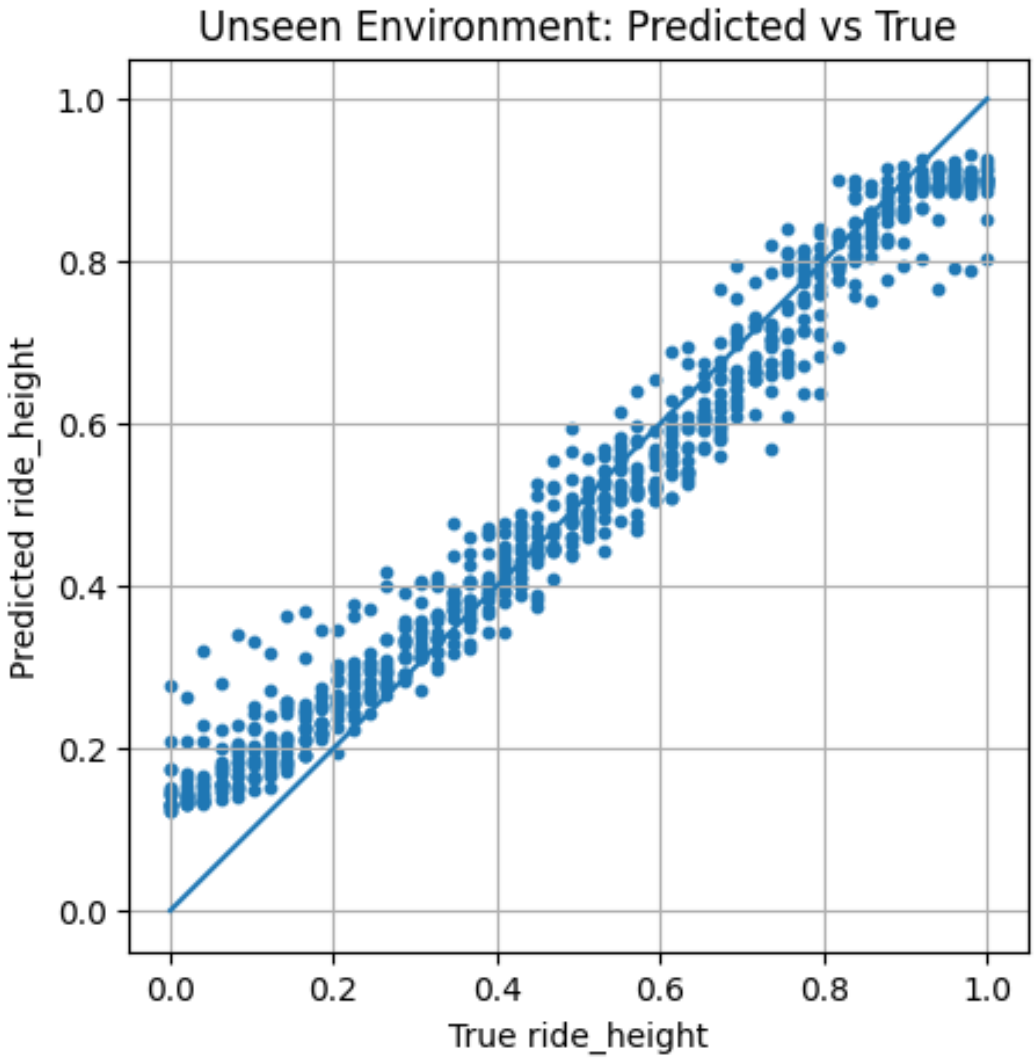

Figure 7

Correlation between true and predicted normalized ride-height values for the custom CNN using data with novel backgrounds.

Figure 8 shows the training history for the custom CNN using Huber loss and mean absolute error. The curves demonstrate that the model generally converged in a stable manner over the training period, with both error metrics decreasing as optimization

progressed. The gap between training and validation performance provides a qualitative indication of generalization behavior. If the validation curve remains close to the training curve, overfitting is less likely; however, a larger separation suggests that the model may be becoming too specialized. Overall, these curves support the claim that the custom CNN could be effective.

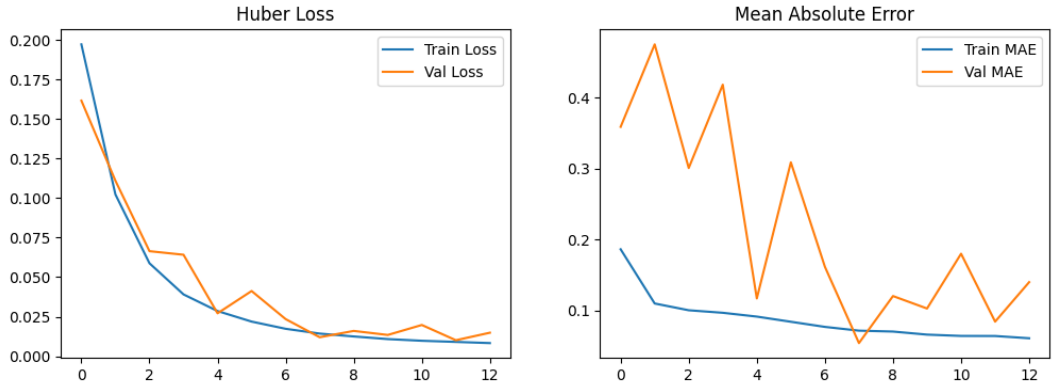

Figure 8

Training curves for Huber loss and mean absolute error (MAE) for the custom CNN.

Custom CNN issues

Even though the custom CNN performed acceptably, the difficulty of designing the model and the time taken to train it are much more complex than just applying ResNet-18 to the problem. There may be advantages to inference time; however, this metric was not tested.

Comparison Between Both Networks

Both model tracks completed end-to-end training on the synthetic dataset and produced stable regression outputs over the tested ride-height range. Qualitatively, predictions followed the expected trend as the hydrofoil body moved relative to the water surface in the rendered frames, indicating that the models learned meaningful visual depth cues rather than random correlations.

Across runs, we observed that generalization was most sensitive to scene diversity rather than raw sample count alone. In particular, variation in HDRI lighting and wake

appearance improved validation stability compared with narrowly repeated conditions. This suggests that synthetic-domain diversity is a key factor for reducing overfitting before transfer to real camera footage.

The ResNet-18 baseline provided a reliable reference and converged predictably. The custom CNN showed competitive behaviour with lower architectural complexity, supporting its usefulness as a deployment-oriented candidate. The remaining error cases were concentrated in frames with heavy spray, ambiguous waterline contrast, or extreme pose combinations that were not yet densely represented in the synthetic distribution.

A direct comparison between the two networks suggests that both models performed strongly when background scenes were similar, but both showed reduced accuracy when evaluated on a diversified, novel scene set. This indicates that the main limitation is not only network architecture, but also how well the synthetic dataset captures environmental variability. Overall, ResNet-18 provided a strong and consistent baseline, while the custom CNN achieved comparable behaviour with lower model complexity, making it a promising option.

Overview of results

Table 3

Model Performance Summary Across Seen and Unseen Environments

Model and Environment	**MAE**	**RMSE**
Custom CNN (seen environment)	0.0538001	0.068332486
Custom CNN (unseen environment)	0.055045385	0.0707691
ResNet-18 (seen environment)	0.015314778	0.019614013
ResNet-18 (unseen environment)	0.015570539	0.02003055

Overall, the results support the feasibility of camera-based ride-height estimation for hydrofoiling craft. However, final conclusions for field use require additional validation on real-world video and domain adaptation to close the synthetic-to-real gap.

Limitations

This project has limitations that must be considered when analyzing overall project success. First, the models were trained and evaluated only on synthetically generated data produced in BlenderProc, rather than on real hydrofoiling videos or images. Although the synthetic dataset includes variation in ride height, lighting, and wake appearance, it cannot fully capture the real-world complexity encountered during operation. Therefore, the reported model performance likely overestimates expected accuracy in the presence of unpredictable spray, reflections, and changing weather conditions.

Moreover, although a diversified dataset was used, it still covers only a limited range of environmental and geometric conditions compared with those encountered by a real hydrofoil craft. Results showed that prediction accuracy decreases when presented with novel background data, suggesting that adding broader and more realistic scene variation may further challenge model performance.

When training the model, it was noticed that the GPU and CPU had very low utilization. This was due to the data decompression of the WebP images. This could potentially be worked around in the future by using a better data pipeline that uses multithreading or some other parallelization technique.

A real-world hardware problem is inference time. It plays a key role, which could lead to prediction delays, making this infeasible for a real-time control system.

Finally, models were compared offline using validation datasets rather than in a real-time system or controlled physical environment, where camera-mount vibration, sensor synchronization, and long-duration stability must also be considered. This limits the predictability of important operational attributes. For this reason, this project should be considered a proof of concept rather than a final sensing method for hydrofoil ride height.

Summary

This project developed and evaluated a computer vision pipeline for estimating hydrofoil ride height from RGB imagery. The workflow combined BlenderProc-based

synthetic data generation with supervised regression training using two model families: a modified ResNet-18 baseline and a custom CNN implemented in TensorFlow/Keras. Results from both same-background and novel-background experiments demonstrated that prediction accuracy is strongly influenced by scene variation, which reinforces the need for synthetic datasets that capture realistic conditions.

The study demonstrates that synthetic training data can be used to learn depth-related visual structure for this application, provided that scene variation is intentionally introduced. Comparative experiments indicate that both architectures can produce consistent predictions, while each offers different trade-offs between robustness and computational efficiency.

These findings establish a practical foundation for future work, including larger-scale data generation, tighter uncertainty analysis, and testing under real operating conditions.

Conclusion

The present work demonstrates that machine-vision regression is a viable approach for estimating hydrofoil ride height without relying solely on traditional ranging sensors. By framing the task as single-value depth prediction from camera images, we obtain a compact and potentially low-cost sensing pathway for high-speed marine environments.

Within the scope of this study, synthetic-data training enabled controlled experimentation and rapid model iteration. The baseline and custom models both captured relevant geometric and water-interface cues, supporting the core project hypothesis. At the same time, observed failure modes under heavy spray and edge-case lighting confirm that additional robustness work is needed before operational deployment. In particular, experiments with novel backgrounds showed that background diversification can reduce regression accuracy, ultimately emphasizing the importance of a wide variety of training data for improving generalization.

Future development should prioritize real-data evaluation, synthetic-to-real adaptation, and integration testing with onboard control systems. With those steps, the

approach can progress from proof-of-concept toward a practical assistive sensing module for hydrofoiling craft.

References

Alzu'bi, A., Al-Ayyoub, M., Jararweh, Y., & Gupta, B. B. (2022). A comprehensive survey on deep learning-based detection and recognition of face masks in the era of the covid-19 pandemic. *Electronics*, *11*(22). Retrieved from https://www.mdpi.com/2079-9292/11/22/3689 doi: 10.3390/electronics11223689

Boatsetter Team. (2024). *What is a hydrofoil boat?* Retrieved from https://www.boatsetter.com/boating-resources/what-is-a-hydrofoil-boat (Accessed March 16, 2026)

Cox, A. (2026, March). *Hydrofoil_CV_Depth_Estimation: Computer Vision for Ride-Height Estimation.* GitHub. Retrieved from https://github.com/AndreCox/Hydrofoil_CV_Depth_Estimation

Denninger, M., Sundermeyer, M., Winkelbauer, D., Zidan, Y., Olefir, D., Elbadrawy, M., ... Durner, M. (2023). Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, *8*(82), 4901. Retrieved from https://joss.theoj.org/papers/10.21105/joss.04901 doi: 10.21105/joss.04901

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition (cvpr)* (pp. 770–778). Retrieved from https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf doi: 10.1109/CVPR.2016.90

Hussain, M. (2023). Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection. *Machines*, *11*(7). Retrieved from https://www.mdpi.com/2075-1702/11/7/677 doi: 10.3390/machines11070677

PyTorch Contributors. (2026). *torchvision.models.resnet18.* Retrieved from https://docs.pytorch.org/vision/main/models/generated/

torchvision.models.resnet18.html

Rak, G., Hocevar, M., Kolbl Repinc, S., Novak, L., & Bizjan, B. (2023). A review on methods for measurement of free water surface. *Sensors*, *23*(4). Retrieved from https://www.mdpi.com/1424-8220/23/4/1842 doi: 10.3390/s23041842

Senix Corporation. (2026a). *Hydrofoil and nautical applications.* https://senix.com/hydrofoil-nautical/. (Accessed: March 16, 2026)

Senix Corporation. (2026b). Toughsonic-100.14 ultrasonic level sensor with 1" npt [Computer software manual]. Lincoln, NE. Retrieved from https://senix.com/toughsonic-100-14-1inch (Accessed: March 16, 2026)

Appendix A

Author Contributions

Table A1

Author Contribution Summary

Author	**Primary Contributions**
Andre R. Cox	Research concept, abstract, synthetic data creation, custom CNN design, editing.
Cara Mikoliunas	Report drafting and editing, and presentation formatting.
Rayman Klar	Baseline model research, ResNet-18 training, report drafting and editing.
Lina Naleto	Introduction, report editing

Appendix B

Concept Diagram

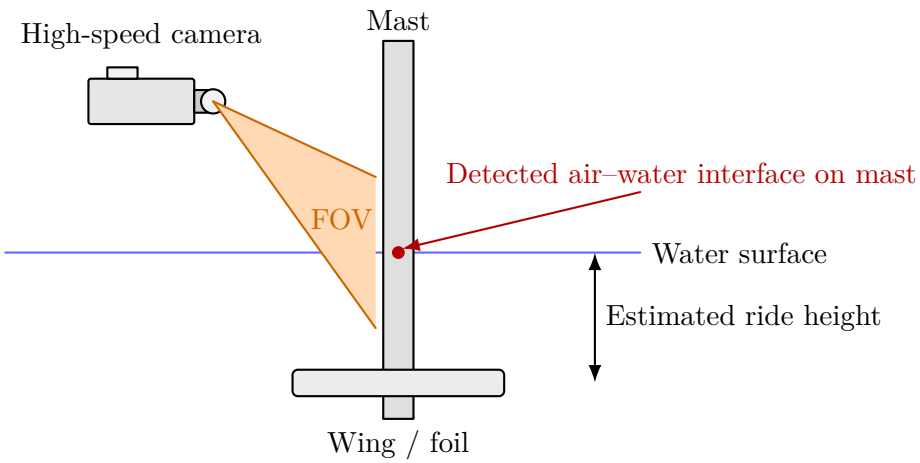

Figure B1

Concept diagram of the proposed vision-based ride-height estimation system.