IJHSR ijhighschoolresearch.org
RESEARCH ARTICLE

Machine Learning-Assisted Airfoil Aerodynamics Performance Prediction

Jason Mak

West Windsor-Plainsboro High School South, 346 Clarksville Rd, Princeton, New Jersey, 08540, USA; jjson.mak@gmail.com

ABSTRACT: Airfoil design plays an important role in determining how efficiently an object travels through the air. It has numerous applications in the industry, ranging from the construction of commercial aircraft to racecars, where small improvements in lift and drag can significantly reduce fuel consumption and improve stability. This study applies machine learning principles to develop a computationally efficient approach for predicting the lift and drag performance of airfoils. Accurate prediction of lift and drag coefficients from computational fluid dynamics software requires computationally expensive solutions of the Navier-Stokes equations. To overcome this limitation, we developed a neural network-based alternative that significantly reduces computational time while maintaining prediction accuracy. First, we generate a training data set with labels using traditional computational fluid dynamics software (CFD). Then we train a convolutional neural network model to make predictions of the aerodynamic coefficients. The results demonstrate that neural networks can effectively learn the complex relationships between airfoil geometry and aerodynamic properties, providing rapid lift and drag predictions not requiring full numerical solution of the Navier-Stokes equations.

KEYWORDS: Physics and Astronomy, Theoretical, Computational and Quantum Physics, Fluid Dynamics, Neural Networks, Airfoils.

Introduction

Neural Networks and Deep Learning:

Machine learning and deep learning are gaining popularity in all disciplines of science and engineering.[1-5] Its cognitive and predictive capability lies in its ability to model and form non-linear responses to complex patterns. Humans refer to such skills as learning from experience. One such area is solving prognostic physics problems, where we set up mathematical equations to solve and predict the evolution of a system and then compute diagnostic information from the state of the system to help us evaluate the model results. This traditional approach is computationally expensive because numerically solving these mathematical equations often requires iteratively solving large matrices through time stepping until satisfactory solutions are found.

At the heart of machine learning is the non-linear neural network, which stores the weights learned during training. We can think of these weights as the experience learnt by the model. A neural network generally consists of three components: the input, the neurons, and the output. The neurons contain weights, biases, and activation functions that are used to transform the input into the output. The input can be thought of as a vector that describes the features of a problem, such as the pixel values of an image, the frequency and sound level values of an audio, etc. Based on the features that capture the main characteristics of the input, the neural network can calculate a set of desired outputs, e.g., the handwritten digit in the image or the mood expressed by the sound.

In the supervised learning approach, the neural network first undergoes a training process. Pre-selected input and output pairs are used to update the weights and biases stored in the neurons until the difference between the given output and the calculated output from the neural network is minimized. This process is similar to what occurs in a gradient descent algorithm. The quality of the training dataset (the input and output pairs) determines the quality of the trained neural network. After the neural network has been trained and verified, we can begin using it to make predictions for inputs that have not been seen by the neural network. Because the neural network calculations during this stage involve the forward multiplication of matrices and vectors, predictions can be made quickly. In the conventional approach, numerical solvers designed to make predictions involve iterative processes, such as the Navier-Stokes equation solver in computational fluid dynamics. This causes the conventional prediction approach to be less efficient and slower than the predictions performed by the trained neural network.[6-8]

A problem that is of interest is to apply such a supervised learning approach to make predictions of airfoil design in aerodynamics quickly.[9,10] Although neural network predictions may not be as precise as dedicated CFD software or results from wind tunnel measurements, they can provide quick feedback on whether a certain design is viable and suggest avenues for improving the design.[11] Such an approach is not limited to the airfoil problem; it has applications in many vision, audio, or signal-based systems, where we can rapidly improve the design by analyzing the image or audio signals.

DOI: 10.36838/IJHSR88.16

Methods

Convolutional Neural Network:

Traditionally, airfoil analysis is done by solving the Navier-Stokes equations in a meshed region around the airfoil with a prescribed free stream inlet and zero pressure gradient outlet boundary condition (Figure 1). The discrete mesh follows the geometry of the airfoil with a carefully generated boundary layer for flow analysis at the surface of the airfoil. The state variables pressure and velocity are calculated in each cell of the mesh during each time step until a quasi-equilibrium solution is obtained. These airfoil calculations are two-dimensional: assuming the wind is in the x-direction and height increases in the z-direction, from the pressure variables, the lift and drag coefficients can be computed by

$$C_l = \hat{z} \cdot \oiint P d\vec{A}$$

$$C_D = \hat{x} \cdot \oiint P d\vec{A}$$

Existing CFD software, such as Simflow and XFOIL, can calculate the lift and drag coefficients at different angles of attack for a given airfoil with reasonable speed on a personal computer.[12,13] Can we design a machine learning model to make predictions of the coefficients after it has been trained? It's a perfect problem where a complex relationship exists between the coefficients and the geometry of the airfoil and the angle of attack (AoA). The CFD models perform these predictions by directly solving the non-linear partial differential equations based on the geometry of the airfoil and the given angle of attack. If such a non-linear relationship exists from (airfoil, AoA) to lift and drag coefficients, the model should be able to learn it! The framework that allows us to learn from the geometry of an airfoil and angle of attack and predict the coefficients is called a convolutional neural network (CNN).[14]

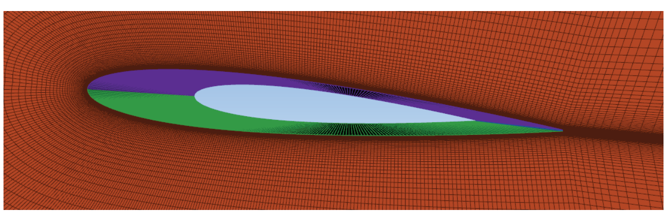

Figure 1: Generated mesh for airfoil NACA 0012 at an angle of attack of 5 degrees. Boundary layer cells are smaller than the cells further away. State variables pressure and velocity are computed in each cell in each time step. Diagnostic variables such as lift and drag coefficient are then computed from the pressure distribution. The green and purple parts of the mesh are on the frontal and rear surfaces of the airfoil.

We constructed a CNN model to analyze the airfoil images and trained the model using coefficients calculated from XFOIL. The convolutional layer can recognize the curvature and the tail of an airfoil through the various kernels and the pooling algorithm used. The convoluted data is then collapsed and concatenated with the angle of attack to form the input vector to the neural network. Figure 2 shows the CNN architecture used for machine learning. The input airfoil image is first convoluted with 32 kernels of size 3 and then max-pooled with a size of 2. Another iteration of convolution with 64 kernels of size 3 and maxpooling of size 2 is applied before the final arrays of images are collapsed with an angle of attack to form the input to the neural network. The neural network has an input layer, a dropout layer of 20%, another linear layer, and finally the output layer. To simplify analysis, we have chosen to focus on the lift coefficient only. Therefore, the output layer is of size 1.

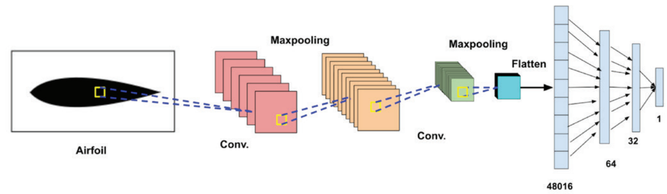

Figure 2: CNN architecture for the airfoil machine learning setup. The input airfoil image is converted into a monochrome black image. It's convoluted and pooled to create 64 channels of input and concatenated with the AoA to train the neural network. The airfoil image after the convolutional layer, plus 16 linear mapping of AoA, becomes 48016 input neurons to the neural network.

We downloaded the airfoil dataset from the official NACA database. The airfoil data are then converted to images of the same size without distortion based on the airfoil data. Figure 3 shows some of the airfoil images used to train the neural network.

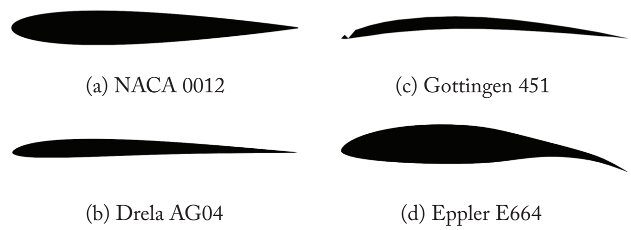

Figure 3: Airfoil images generated from the NACA airfoil database.[15] Occasionally, the image generated from the airfoil data can be defective, as shown in (c) for the airfoil Gottingen 451. This could be a result of polygon self-intersection when there is an error in the coordinates. These images are checked and eliminated from the training dataset. The airfoils with a round leading edge are usually subsonic as in (a), (b), and (d).

Setting up XFOIL and Simflow:

To verify the aerodynamics calculation and prepare training data for the CNN, we set up XFOIL and Simflow to run under similar conditions. XFOIL is designed specifically for such simulations and is faster to run. Simflow provides an airfoil mode for aerodynamics calculations and is used to verify the results calculated from XFOIL. Table 1 shows the lift coefficients calculated from XFOIL and Simflow for the airfoil ag04 at various angles of attack. The results from XFOIL and Simflow are in good agreement.

Table 1: Comparison of calculated lift coefficients between XFOIL and Simflow for the AG04 airfoil at various angles of attack. As the angle of attack increases, we can see a monotonic increase in the lift coefficients for both XFOIL and Simflow. This agrees with the physics of airfoils for AoAs less than the stall angle, as with a higher angle of attack, more air will be deflected downwards—creating a higher pressure difference/lift.

Angle of Attack	XFOIL	Simflow
0.0	0.1764	0.18
1.0	0.2916	0.29
2.0	0.4068	0.40
3.0	0.5219	0.50
4.0	0.6368	0.61
5.0	0.7514	0.72
6.0	0.8659	0.82

DOI: 10.36838/IJHSR88.16

During training, two NACA airfoils, GOE 451 and Marsden, were deleted from the NN training sample due to abnormally high lift coefficients given by Simflow and XFOIL. In both XFOIL and Simflow, we have used an incompressible steady-state solver. XFOIL does not provide many options in fine-tuning the calculation because it is specifically designed for such calculations. For Simflow, we have chosen the SIMPLE solver with a Reynolds-averaged Navier-Stokes (RANS) turbulent scheme. Simflow also allows for fine-tuning of the generated mesh with boundary layer cell thickness and length. We used the default settings from Simflow for the airfoil calculation. The airfoil radius is left as the default value of 20 meters, and a Mach number of 0.3 or 100 m/s is used as the free stream air velocity. The default kinematic viscosity of air, 1.51 * 10-5 m^2/s, is used in the viscous calculation. This is equivalent to a Reynolds number of

$$\frac{uL}{\nu} = \frac{100\, m/s \cdot 20\, m}{1.51 \cdot 10^{-5} m^2/s} = 1.32 \cdot 10^{8}$$

In theory, the lift coefficient does not depend on the Mach number; it should be a function of geometry and angle of attack only. In practice, the calculated lift coefficients from XFOIL and Simflow show dependencies on the Reynolds number used.

Results and Discussion

We trained the CNN model with 173 images, each for 21 angles of attack (from -10.0 to +10.0 degrees with a 1.0 degree interval), after discarding many airfoils from UIUC due to numerical convergence issues at various angles of attack in the XFOIL calculations. Most of the images used to train CNN have a round leading edge, which usually indicates subsonic flight. The training data for each input configuration are the lifting coefficients calculated using the XFOIL program.[16] We also ran Simflow to validate the results calculated from XFOIL. A total record of 3633 entries in the format (airfoil image, angle of attack, lift coefficient) is used to train the CNN. After the CNN is trained, we use the model to make predictions of airfoils at various angles of attack. The test data set includes approximately 200 airfoils chosen randomly from the UIUC database with an SRS. A comparison of the two sets of coefficients is shown in Figure 4. The predicted coefficients are generally in agreement with the corresponding coefficients calculated from XFOIL, with a correlation coefficient squared of 0.997, indicating excellent agreement between the machine learning predictions and traditional CFD methods. To quantify the benefit of this approach, for an airfoil set at a specific angle, traditional CFD simulations like Simflow average several minutes in order to reach numerical convergence. In contrast, the trained CNN required under 30 seconds to predict the lift coefficients for the entire test data set at the given angles of attack, corresponding to millisecond-level inference per case.

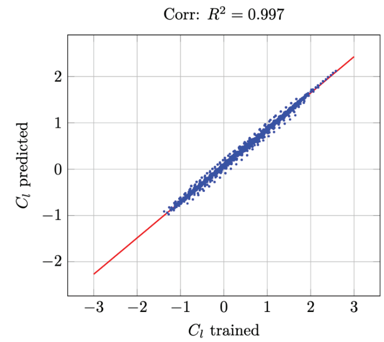

Figure 4: Comparison of predicted lift coefficients from XFOIL calculations and those obtained from the trained neural network across various airfoils from the UIUC database and different angles of attack. There is a clear linear trend with a best-fit line of y=0.783x+0.08. This trend is similar to that of the line y=x, representing perfect predictions of the lift coefficient.

Despite the high correlational coefficient, the CNN model exhibits a consistent systematic bias toward underestimating lift coefficients compared to XFOIL calculations. The regression analysis reveals a trend line slope of 0.783 with a y-intercept of 0.08. indicating that the model predictions are consistently lower than the reference values by approximately 22%. This underestimation trend is particularly evident across different airfoil geometries and angle of attack ranges, suggesting that the model may be conservative in its predictions. The systematic nature of this bias indicates that it could potentially be corrected through adjustment of the neural network architecture, though the underlying physical mechanisms causing this bias require further investigation.

The validation approach employed both XFOIL and Simflow calculations to ensure robustness of the reference data, with both tools configured for incompressible steady-state analysis at a Reynolds number around 1.32 * 10^8. While XFOIL and Simflow generally showed good agreement for most test cases, some discrepancies were observed, particularly for certain airfoil geometries. As shown in Table 1, the calculated results for the AG04 airfoil showed excellent agreement between XFOIL and Simflow across all angles of attack, with differences typically less than 5%. However, for other airfoils such as NACA 0015 and BW 050209, larger discrepancies were observed between the two CFD tools, highlighting the inherent uncertainties in numerical simulations and the importance of using multiple validation sources when training machine learning models for aerodynamic predictions.

DOI: 10.36838/IJHSR88.16

Table 2: Comparison of lift coefficients (Cl) from XFOIL, Simflow, and the trained neural network across various airfoils and different angles of attack. For this set of airfoils, the predicted values tend to be larger than the Simflow and XFOIL values; however, if we take into account the hundreds studied, there is a general trend of underestimation.

AoA	XFOIL	Simflow	Predicted	AoA	XFOIL	Simflow	Predicted
1.0	0.1163	0.223	0.2152	1.0	-0.0245	0.067	0.2132
2.0	0.2364	0.327	0.2936	2.0	0.0983	0.163	0.3000
3.0	0.3565	0.427	0.4137	3.0	0.2211	0.274	0.4288
4.0	0.4764	0.543	0.5051	4.0	0.3439	0.376	0.5216
5.0	0.5962	0.650	0.5934	5.0	0.4666	0.465	0.6098

(a) b737b (b) naca0015

AoA	XFOIL	Simflow	Predicted	AoA	XFOIL	Simflow	Predicted
1.0	0.5461	0.553	0.7424	1.0	0.2918	0.132	0.2876
2.0	0.6674	0.683	0.8527	2.0	0.4194	0.235	0.4256
3.0	0.7884	0.691	0.9606	3.0	0.5469	0.340	0.5212
4.0	0.9092	0.907	1.0651	4.0	0.6743	0.4407	0.6158
5.0	1.0297	0.924	1.1626	5.0	0.8014	0.538	0.7068

(c) goe701 (d) bw050209

AoA	XFOIL	Simflow	Predicted	AoA	XFOIL	Simflow	Predicted
1.0	0.3346	0.292	0.3359	1.0	0.6773	0.572	0.5897
2.0	0.4516	0.395	0.4036	2.0	0.795	0.671	0.6909
3.0	0.5685	0.496	0.4974	3.0	0.9126	0.768	0.7986
4.0	0.6852	0.594	0.5921	4.0	1.0298	0.861	0.9032
5.0	0.8017	0.689	0.6806	5.0	1.1467	0.946	1.0006

(e) rg14 (f) mh116

The model's performance varies significantly across different airfoil geometries, as illustrated in the comparative analysis (Table 2, Figure 5) of six representative airfoils. While some airfoils, like B737 B and RG 14, show relatively good agreement between predicted and calculated values, others exhibit more substantial deviations. The GOE 701 airfoil demonstrates the model's tendency to overestimate lift coefficients for high-lift airfoils. In contrast, symmetric airfoils like NACA 0015 show the model's difficulty in capturing the near-zero lift coefficients at low angles of attack. These airfoil-specific variations suggest that the current CNN architecture may benefit from additional geometric feature extraction layers to improve prediction accuracy across the full spectrum of airfoil designs.

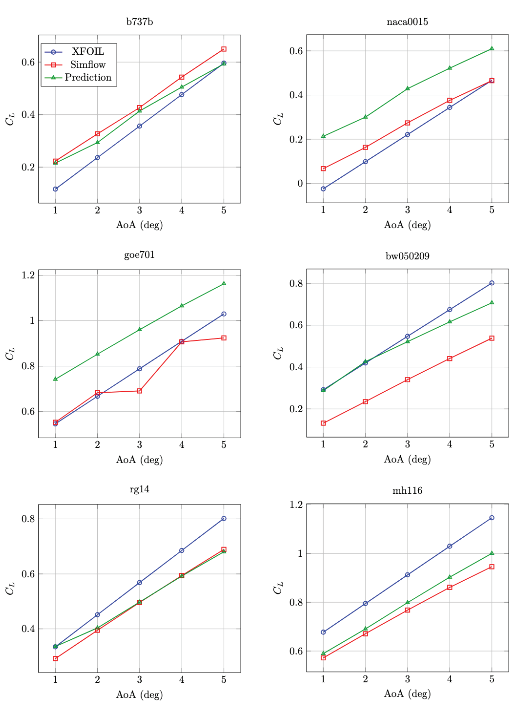

Figure 5: Comparison of lift coefficients obtained from XFOIL calculations (blue), Simflow calculations (red), and CNN predictions (green). The graphs indicate a linear trend except for some outliers (e.g., Simflow for GOE 701), suggesting a positive relationship between the lift coefficient and the angle of attack, which is physically true until the airfoil reaches the stall angle, where the lift starts to decrease.

Conclusion

In this work, we compare the lift coefficient predictions of a trained neural network against numerical calculations from XFOIL and Simflow using airfoils from the UIUC database over a range of angles of attack. The results demonstrated that the neural network can capture the overall trends in aerodynamic performance, often following the correct slope and relative magnitude of the lift curves. However, in several cases, the predicted values underestimate the reference data, as demonstrated by the scatterplot comparison in Figure 4. This underestimation highlights both the promise of the model and the need for further refinement to improve quantitative accuracy.

A key aspect of improving the model involves error analysis and mitigation. One immediate adjustment is to significantly increase the number of training epochs, as the relatively high validation error suggests that the network has not yet converged to an optimal solution. This could directly reduce the systematic underestimation observed in Figure 4. Additionally, adjusting the CNN model by adding more convolutional layers and changing the neural network layout could help improve the results.

Finally, the robustness of the results depends not only on model training but also on data integrity. Since visual inspection of CFD images or intermediate representations for errors is challenging, an automated verification mechanism may be required. One possible solution would be to train an auxiliary CNN tasked specifically with detecting corrupted or misaligned input images, thereby ensuring higher data quality for the main prediction network. While the present results are encouraging in demonstrating the feasibility of aerodynamic prediction with machine learning, careful attention to training strategies, model setup, validation practices, and data quality control will be essential for pushing the model toward application-ready performance.

Acknowledgments

I would like to express my gratitude to my mentor, Dr Fei Liu, for guiding me through this research and answering the many questions I had.

References

1. McCulloch, W. S.; Pitts, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133.
2. Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychol. Rev.* **1958**, *65*, 386–408.
3. Minsky, M.; Papert, S. *Perceptrons: An Introduction to Computational Geometry*; MIT Press: Cambridge, MA, **1969**.
4. Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning Representations by Back-Propagating Errors. *Nature* **1986**, *323*, 533–536.
5. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
6. Kochkov, D.; Smith, J.; Alieva, A.; Wang, Q.; Brenner, M. P.; Hoyer, S. Machine Learning–Accelerated Computational Fluid Dynamics. *Proc. Natl. Acad. Sci. U.S.A.* **2021**, *118*, e2101784118.

7. Vinuesa, R.; Brunton, S. L. Enhancing Computational Fluid Dynamics with Machine Learning. *Nat. Comput. Sci*. **2022**, *2*, 358–366.
8. Sharma, P.; Chung, W. T.; Akoush, B.; Ihme, M. A Review of Physics-Informed Machine Learning in Fluid Mechanics. *Energies* **2023**, *16*(5), 2343.
9. Li, J.; Du, X.; Martins, J. R. R. A. Machine Learning in Aerodynamic Shape Optimization. *Prog. Aerosp. Sci*. **2022**, *134*, 100849.
10. Loh, P. D.; Ong, O. A. Reinforcement Learning Approach to Airfoil Shape Optimization. *Sci. Rep*. **2023**, *13*, 9753.
11. Deters, R. W.; Williamson, G. A.; Selig, M. S. Wind Tunnel Testing Airfoils at Low Reynolds Numbers. In *49th AIAA Aerospace Sciences Meeting*; American Institute of Aeronautics and Astronautics: Orlando, FL, **2011**; Paper AIAA 2011-875.
12. *SimFlow*, Ver. 5.0; SimFlow Technologies, 2025.
13. Drela, M. XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils. In *Low Reynolds Number Aerodynamics*, Mueller, T. J., Ed.; Springer: Berlin, Heidelberg, **1989**; pp 1–12.
14. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324.
15. *UIUC Airfoil Coordinates Database*; Selig, M. S.; University of Illinois at Urbana-Champaign. https://m-selig.ae.illinois.edu/ads/coord_database.html
16. Drela, M.; Youngren, H. XFOIL, Ver. 6.99; Massachusetts Institute of Technology, 2013; http://web.mit.edu/drela/Public/web/xfoil/

Author

Jason Mak is a rising senior at West Windsor-Plainsboro High School South who enjoys learning physics and math. He hopes to pursue physics, engineering, or astronomy in college. In his free time, he plays badminton, tries new foods, stargazes, and reads the occasional book.

DOI: 10.36838/IJHSR88.16