Jurnal Polimesin
Department of Mechanical Engineering
Lhokseumawe State Polytechnic
http://e-jurnal.pnl.ac.id/polimesin

e-ISSN	: 2549-1999	No	: 2	Month	: August
p-ISSN	: 1693-5462	Volume	: 20	Year	: 2022

Article Info

Received	: 2022-09-17	Accepted	: 2022-08-20
Revised	: 2022-08-20	Available online	: 2022-08-31

Neural Network Approach for Predicting Aerodynamic Performance of NACA Airfoil at Low Reynolds Number

Mohamad Yamin*, Zaid Alkahfi Raamadhan
Department of Mechanical Engineering, University of Gunadharma, Depok, 16424, Indonesia
***** Corresponding Author: mohay@staff.gunadarma.ac.id

Abstract

In designing and developing airfoils, confirmation of proper design performance under various flow conditions is vital. Experimental studies using wind tunnels or numerical simulations can often utilize. In some cases, numerical studies have a weakness in computational time. This study focuses on predicting the drag coefficient of the airfoil using the CNN machine learning architecture. Starting with a numerical simulation of 500 types of NACA airfoils with a Reynolds number of 4000 using XLRF5 software to obtain image data, lift and drag coefficients. The training, test, and validation dataset uses numerical simulation results as labels. ReLU is the activation function used in this study, with Adam optimizer and MSE loss function. It achieved a relative error of 8% in predicting the drag coefficient. With the results obtained, aircraft designers can use the method to predict the drag coefficient value from various geometries.

Keywords:
Airfoil NACA, Supervised learning, drag prediction, CNN, XLRF5

1 Introduction

Machine Learning (ML) [1] and Artificial Intelligence (AI) [2] are currently in great demand in several scientific fields, for example, in the aerospace industry [3]. The aerospace industry is constantly looking for efficient ways to accelerate development to meet increasing demands and achieve superior components. Air traffic management [4], operational efficiency improvement [5], product design [6], customer service [7], pilot training [8], and research [9]-[11] are all areas where artificial intelligence can be used.

Knowing the phenomenon in a flow can be done in two ways. The first is to use an experimental approach[12]. Then the second is to use numerical studies, commonly known as CFD [13]. However, empirical studies and numerical simulations still have drawbacks in terms of cost and time. Therefore, Machine Learning was introduced to save time and cost.

Aerodynamic modeling based on the CFD [14] approach is now an essential tool for designing an aircraft [13]. Selecting the right airfoil design is needed to get an efficient method. Over the past decade, computing resources have increased rapidly, but CFD simulations (DNS, RANS, and LES [15]) require computational effort. Artificial intelligence and especially Machine Learning can be involved in handling a large amount of data to extract knowledge from the data involved. Machine Learning offers a flexible framework for solving a variety of fluid mechanics issues, such as shape optimization, reduced-order modeling, experimental processing, turbulence closure, and control. Machine Learning automates tasks and augments human domain knowledge.

Convolutional neural networks were first proposed [16] for use in image processing[17]. CNN can collect feature points from based data (such as images) by sliding the trainable kernel across the image and can perform convolutions on every pixel.

This research investigates machine learning techniques and applies aerodynamic problems predicting drag and lift coefficients for different airfoils.

2 Research Methods

This section will explain collecting datasets and the Machine Learning methods used.

2.1 Dataset Construction

The wing's cross-sectional form in two dimensions is an airfoil [18] and is typically described by the geometric parameters defined in Fig. 1. The chord line is the horizontal line that connects the leading and trailing edges [19]. The mean-camber line governs the airfoil's curvature between the upper and lower surfaces. When the mean-camber and chord-line of an airfoil intersect, the airfoil is said to be asymmetric. In addition, The maximum thickness is another important parameter that describes the airfoil geometry. Maximum thickness value and distance from the leading edge are usually expressed as a percentage of the airfoil's chord-line length.

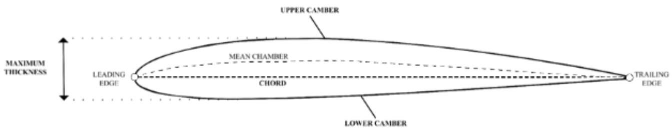

Fig. 1. A typical airfoil geometry

In this study, 500 different naca geometries of airfoils were chosen. An example of the selected airfoil, the geometry shown in Fig. 2, is a NACA airfoil [20] with a series of 4 and 5 digits.

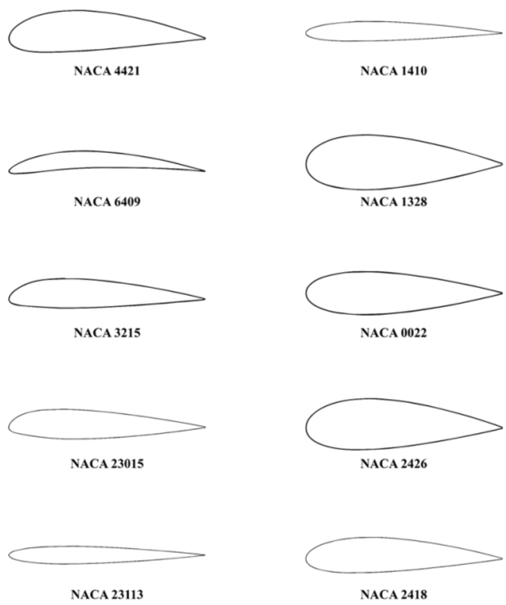

Fig. 2. Airfoil data train geometry

2.2 Simulation

The inviscid method analyzed selected airfoils in XFLR5 [21] software. Inviscid XFLR5 analysis in two dimensions has a linear vorticity flow function formulation. The program generates a 2-D inviscid airfoil flowfield for analysis. This flowfield comprises both freestream flow and vortex sheet on the airfoil. The flow function can be stated as [22]

$$\psi(x,y) = u_\infty - v_\infty + \frac{1}{2\pi}\int \gamma(s)\ \mathrm{lnr}\ (s;x,y) + \frac{1}{2\pi}\int \sigma(s)\theta(s;x,y)\ ds \qquad (1)$$

Where,
σ = the Power of the source sheet,
γ = Eddy sheet strength,
s = the coordinates through Eddy and the source sheet,
$u_\infty; v_\infty$ = the component of the free flow velocity,
r = the vector magnitude between the field points x, y, and the point s, and
θ = the angle vector.

The Eppler E387 airfoil was analyzed under the same conditions as the results of the wind tunnel experiment on Langley's low-turbulence pressure to verify the XFLR5 computation [23]. Similar to the experimental investigation by Morgado [24], XFLR5 analysis was performed on Reynolds number $2x10^5$ and Mach number 0.06 utilizing greater than 150-panel counts to design an airfoil. The results were not significantly different.

Moving an object through a fluid produces two forces known as the drag and lift forces. While lift force is perpendicular to the direction of the relative airflow, a drag force is a parallel. The general lift and drag formulas are expressed in the equation 2 to 3

$$L = \frac{1}{2}\rho A V^2 C_L \qquad (2)$$

$$D = \frac{1}{2}\rho A V^2 C_D \qquad (3)$$

In this study, XFLR5 analysis was carried out on the Reynolds number 4×10^3 with chord length value one and variation angle of attack from 0° to 15°. The simulation result data is stored in a CSV file as a dataset for the Machine Learning process.

2.3 Machine Learning Method

Viquerat [25] developed the machine learning approach used in this study to estimate the drag force on any 2D plane. This approach begins with data preparation or preliminary processing of data. Data pre-processing is an essential component of an artificial neural network since it substantially impacts the network's ability to learn. The input is a pixel-based, single-channel picture (white and black images). The value of the pixel range is 0 to 255. These images are reduced to pixels during the pre-processing phase, and their values are scaled between 0 and 1. This normalization is essential because gradient descent can be prevented in convergent networks by applying the back-propagation method with large and heterogeneous values (Fig. 4) [26].

The input dataset is separated into three sections. First, a training set, where learning will take place; second, a validation set, which is used to periodically assess network correctness throughout training; and finally, a test set, where the network's ultimate performance is evaluated [27]. While not strictly relevant to the current issue, comments about the possibility of using data augmentation should be made. When a slight transformation of the input image should not change the associated label (as is the case for most classification purposes), the available data set can be artificially increased by including the transformed input image associated with the original label. This CNN baseline drag prediction network consists of two layers, as shown in blue, and a max-pooling layer, as shown in yellow. When the pattern is repeated five times, the image size is reduced by one-half.

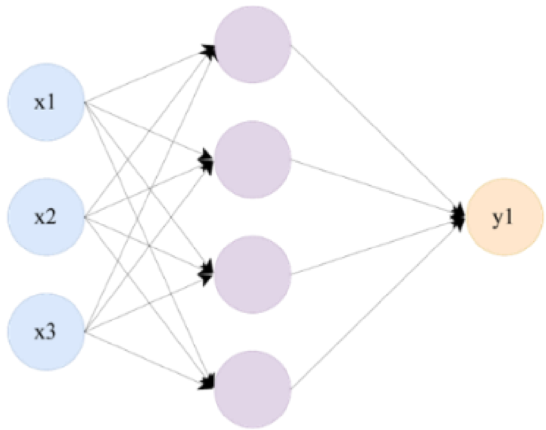

Fig. 4. A convolutional network for drag prediction

The activation function produces a nonlinear transformation between the input and output spaces [28]. Usually, they are chosen by layer. In classification cases, sigmoid or hyperbolic tangents are typically utilized. ReLU, or the activation function of rectified linear units[29], is an excellent choice for regression.

The learning process of a neural network involves modifying the network's biases and weights to decrease the value of a loss function that has been carefully chosen. Weights and biases are optimized using stochastic gradient descent (SGD). This approach, which uses the chain rule to calculate how much each network's degrees of freedom contribute to the loss value, is fundamental to the learning process, as shows Fig. 5.

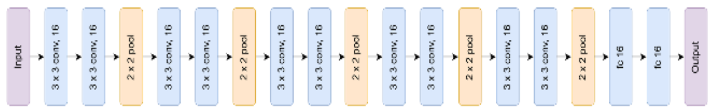

Fig. 5. The training function and validation loss

Two fully connected layers immediately succeed the last layer with maximum pooling (in magenta). It concludes with a one-dimensional layer whose linear activation function outputs the requisite resistance.

The network is regularly exposed to specific input data during the training phase. Epochs are complete iterations across all input samples, and advanced networks frequently train for thousands of epochs. The number of training epochs in this study was set to 500. The network has been exposed to a random set of data, after which the SGD step is performed.

The dimension of a neural network, measured by the number of layers and neurons inside each layer, is an essential factor to consider while designing one. Some network that is too small for the task will be unable to comprehend the complexities of the implicit function that must be mapped. On the other hand, a network with an excessive number of degrees of freedom will overfit, which prevents it from generalizing to new data because it fits the training set too well.

There are several ways to prevent overfitting. The first is to acquire more data. However, this is frequently impractical or expensive. The second alternative is to scale back the network so that the independent parameters are proportional to the amount of raw data available. It might result in a loss of generalizability. The final decision is to use regularization. Utilizing early stop[30], which typically entails monitoring validation set loss improvements throughout training and terminating training when these losses stop improving, can also help avoid overfitting. Extra parameters prevent premature learning termination and can be defined as postponing the first halt by a specific time.

Recently, more pre-made neural network libraries have been available, mainly using Python [32] or C++ [31]. Typically, supervised learning techniques offer a range of activation functions, layer types, losses, and optimizer options.

Because of the high level of abstraction and user-friendliness offered by the Python programming language, we decided to use Keras [26, 33] (with the Tensorflow backend [34, [35]). The generic convolutional network design shown in Fig. 4 constructed using the fundamental Keras layer is considered in the remaining sections of this essay. The convolution pattern is repeated in this network, with various filters for each layer. The template has two convolution layers, with the first layer having eight filters. Two dense layers of size 16 are used to terminate the network. While all other network levels use ReLU activation, the final layer uses linear activation functions and shows predictions.

3 Results and Discussion.

Section 2.3 on drag prediction introduces this part's primary network performance evaluation. The input image size was 255x255, and there were 66.497 learnable parameters. The learning rate was set at $1{\times}10^{3}$, the decay factor at $5{\times}10^{3}$, and early breaking was used to decide when the training was completed. Without prior training, network parameters are determined arbitrarily, with a default batch size of 64. One training epoch takes around 25 seconds to end on an RTX 2080 GPU card for 350 seconds. In this study, Adam functions as the optimizer while MSE functions as the loss. Fig. 6 depicts validation and loss graphs.

Fig. 6. Training function and loss of validity

In determining the predictive performance of the network, the relative resistance prediction error in the test subset is computed. For each subset form, a forward network path is generated in order to compare the projected bottleneck to the actual blockage. Following that, The relative prediction error is calculated. Fig. 7 displays a plot of the test set's error rates. A low relative error indicates good overall accuracy.

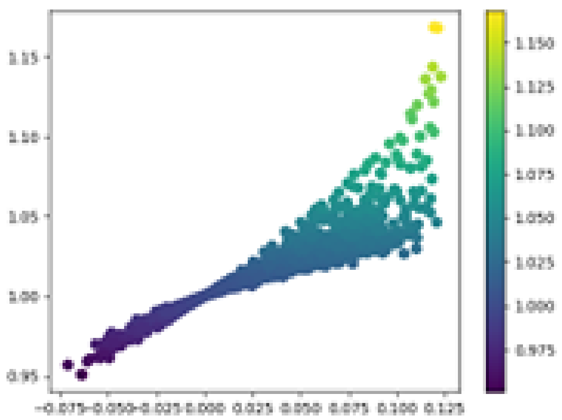

Fig. 7. The relative error in drag prediction

This study developed a convolutional neural network to forecast drag at Re $4x10^{3}$. The network is trained using 500 NACA airfoil forms from a proprietary data set, and the drag force is derived numerically by solving the Navier-Stokes equation.

The data set's large variety of geometric shapes enables the network to forecast drag accurately, with a maximum relative error in the 8- percent range. Table 1 shows the predicted drag and its comparison with the exact drag and relative error.

Table 1. Prediction, error, and exact drag airfoil NACA

Shapes	Description	Prediction (Relative error)	Exact drag
	NACA 0012	0.12493233 (8.3%)	0.11535
	NACA 2415	0.05851308 (6.87%)	0.05475
	NACA 4414	0.09124382 (4.81%)	0.09586
	NACA 6412	0.13576797 (8.81%)	0.1489
	NACA 4424	0.10406933 (6.65%)	0.11149
	NACA 1412	0.1606595 (3.55%)	0.16009
	NACA 0024	0.07214748 (7.05%)	0.07057
	NACA 2408	0.20848563 (7.77%)	0.19345
	NACA 22112	0.12015762 (6.28%)	0.18965
	NACA 24112	0.20415007 (6.13%)	0.19237

4 Conclusions.

Open-source software XLRF5 was used for generating the dataset. Five hundred types of NACA airfoils were numerically simulated to find coefficient lift (cl) and drag (cd) values with various angles of attack from 0° to 15°. The presented convolutional neural network algorithm could predict the drag coefficient of the NACA airfoil using the training dataset. From the prediction results obtained a top cd relative error of 8%.

References.

[1] Subasi, "Chapter 3 - Machine learning techniques," A. B. T.-P. M. L. for D. A. U. P. Subasi, Ed. Academic Press, 2020, pp. 91–202.

[2] P. H. Winston, Artificial intelligence. Addison-Wesley Longman Publishing Co., Inc., 1992.

[3] L. Zhu, W. Zhang, J. Kou, and Y. Liu, "Machine learning methods for turbulence modeling in subsonic flows around airfoils," Phys. Fluids, vol. 31, no. 1, p. 15105, 2019.

[4] A. Ouallane, A. Bahnasse, A. Bakali, and M. Talea, "Overview of Road Traffic Management Solutions based on IoT and AI," Procedia Comput. Sci., vol. 198, pp. 518–523, 2022, doi: https://doi.org/10.1016/j.procs.2021.12.279.

[5] M. Doumpos, C. Zopounidis, D. Gounopoulos, E. Platanakis, and W. Zhang, "Operational research and artificial intelligence methods in banking," Eur. J. Oper. Res., 2022, doi: https://doi.org/10.1016/j.ejor.2022.04.027.

[6] Krahe, M. Iberl, A. Jacob, and G. Lanza, "AI-based Computer Aided Engineering for automated product design - A first approach with a Multi-View based classification," Procedia CIRP, vol. 86, pp. 104–109, 2019, doi: https://doi.org/10.1016/j.procir.2020.01.038

[7] M.-H. Huang and R. T. Rust, "Artificial intelligence in service," J. Serv. Res., vol. 21, no. 2, pp. 155–172, 2018.

[8] N. Ahmadi, M. Romoser, and C. Salmon, "Improving the tactical scanning of student pilots: A gaze-based training intervention for transition from visual flight into instrument meteorological conditions," Appl. Ergon., vol. 100, p. 103642, 2022, doi: https://doi.org/10.1016/j.apergo.2021.103642.

[9] H. Zhu, "Research on intelligent analysis strategies to improve athletes' psychological experience in the era of artificial intelligence," Prog. Neuro-Psychopharmacology Biol. Psychiatry, p. 110597, 2022, doi: https://doi.org/10.1016/j.pnpbp.2022.110597

[10] Y. P. Tsang and C. K. M. Lee, "Artificial intelligence in industrial design: A semi-automated literature survey," Eng. Appl. Artif. Intell., vol. 112, p. 104884, 2022, doi: https://doi.org/10.1016/j.engappai.2022.104884.

[11] K. K. Ramachandran, A. Apsara Saleth Mary, S. Hawladar, D. Asokk, B. Bhaskar, and J. R. Pitroda, "Machine learning and role of artificial intelligence in optimizing work performance and employee behavior," Mater. Today Proc., vol. 51, pp. 2327–2331, 2022, doi: https://doi.org/10.1016/j.matpr.2021.11.544.

[12] M. Desai, S. a Channiwala, and H. J. Nagarsheth, "Experimental and Computational Aerodynamic Investigations of a Car," October, vol. 3, no. 4, pp. 359–368, 2008.

[13] Jameson, "Optimum aerodynamic design using CFD and control theory," in 12th computational fluid dynamics conference, 1995, p. 1729.

[14] X. Wang, J. Kou, and W. Zhang, "Unsteady aerodynamic modeling based on fuzzy scalar radial basis function neural networks," Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng., vol. 233, no. 14, pp. 5107–5121, 2019.

[15] Celik, "RANS/LES/DES/DNS: The future prospects of turbulence modeling." 2005.

[16] H.-J. Yoo, "Deep convolution neural networks in computer vision: a review," IEIE Trans. Smart Process. Comput., vol. 4, no. 1, pp. 35–43, 2015.

[17] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," Handb. brain theory neural networks, vol. 3361, no. 10, p. 1995, 1995.

[18] R. L. Fearn, "Airfoil aerodynamics using panel methods," Math. J., vol. 10, no. 4, p. 15, 2008.

[19] İ. H. GÜZELBEY, Y. ERASLAN, and M. H. DOĞRU, "Numerical investigation of different airfoils at low Reynolds number in terms of aerodynamic performance of sailplanes by using XFLR5," Karadeniz Fen Bilim. Derg., vol. 8, no. 1, pp. 47–65, 2018.

[20] H. Abbott, A. E. Von Doenhoff, and L. Stivers Jr, "Summary of airfoil data," 1945.

[21] F. Škultéty, B. Badánik, M. Bartoš, and B. Kandera, "Design of Controllable Unmanned Rescue Parachute Wing," Transp. Res. Procedia, vol. 35, pp. 220–229, 2018, doi: https://doi.org/10.1016/j.trpro.2018.12.026.

[22] Gharali and D. A. Johnson, "Dynamic stall simulation of a pitching airfoil under unsteady freestream velocity," J. Fluids Struct., vol. 42, pp. 228–244, 2013, doi: https://doi.org/10.1016/j.jfluidstructs.2013.05.005.

[23] R. J. McGhee, Experimental results for the Eppler 387 airfoil at low Reynolds numbers in the Langley low-turbulence pressure tunnel, vol. 4062. National Aeronautics and Space Administration, Scientific and Technical …, 1988.

[24] Morgado, R. Vizinho, M. A. R. Silvestre, and J. C. Páscoa, "XFOIL vs CFD performance predictions for high lift low Reynolds number airfoils," Aerosp. Sci. Technol., vol. 52, pp. 207–214, 2016.

[25] Viquerat and E. Hachem, "A supervised neural network for drag prediction of arbitrary 2D shapes in laminar flows at low Reynolds number," Comput. Fluids, vol. 210, 2020, doi: 10.1016/j.compfluid.2020.104645.

[26] F. Chollet, Deep learning with Python. Simon and Schuster, 2021.

[27] Mahesh, "Machine learning algorithms-a review," Int. J. Sci. Res. (IJSR).[Internet], vol. 9, pp. 381–386, 2020.

[28] O. Sharma, "A new activation function for deep neural network," in 2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon), 2019, pp. 84–86.

[29] S. Mastromichalakis, "ALReLU: A different approach on Leaky ReLU activation function to improve Neural Networks Performance," arXiv Prepr. arXiv2012.07564, 2020.

[30] R. Caruana, S. Lawrence, and C. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," Adv. Neural Inf. Process. Syst., vol. 13, 2000.

[31] N. M. Josuttis, "The C++ standard library: a tutorial and reference," 2012.

[32] W. Python, "Python," Python Releases Wind., vol. 24, 2021.

[33] F. Chollet, Deep learning mit python und keras: das praxis-handbuch vom entwickler der keras-bibliothek. MITP-Verlags GmbH & Co. KG, 2018.

[34] N. Hasabnis, "Auto-tuning TensorFlow threading model for CPU backend," in 2018 IEEE/ACM Machine Learning in HPC Environments (MLHPC), 2018, pp. 14–25.

[35] Nandy and M. Biswas, "Reinforcement learning with keras, tensorflow, and chainerrl," in Reinforcement Learning, Springer, 2018, pp. 129–15