Aerodynamic Coefficient Prediction of Airfoils by Using Convolutional Neural Networks

Arizal Akbar Zikri,[1, 2, a)] Wahyu Hidayat,[b)] and Acep Purqon[c)]

[1)]*Department of Computational Science, Faculty of Mathematics and Natural Sciences, Bandung Insitute of Technology, Bandung, West Java, Indonesia*

[2)]*National Laboratory for Aerodynamics, Aeroelastics, and Aeroacoustics Technology, National Research and Innovation Agency, South Tangerang, Banten, Indonesia*

[a)]*Corresponding author: ariz002@brin.go.id*

[b)]*Electronic mail: wahid@fi.itb.ac.id*

[c)]*Electronic mail: acep.purqon@itb.ac.id*

Abstract. Many applications use symmetric or asymmetric airfoils, such as aircraft design, wind turbines, and heat transfer. Each airfoil has different aerodynamic coefficients. The aerodynamic coefficients of these airfoils must be obtained to optimize the airfoil design. Engineers use various methods to obtain the aerodynamic coefficients. The prediction method is an approximation method that effectively reduces time and cost. This paper uses a prediction method to get aerodynamic coefficients of airfoils for lift, drag, and pitch-moment using convolutional neural networks (CNN). In CNN, airfoil geometry representation is essential. Airfoil geometries were transformed into grayscale and RGB images using the signed distance function (SDF) and mesh algorithm. Each airfoil representation was trained and tested to generate each prediction model using modified LeNet-5. Simulation results show that the proposed method using a meshing algorithm based on CNN could predict the lift, drag, and pitch-moment coefficient of the airfoil with varying angles of attack simultaneously.

INTRODUCTION

The airfoil design continues to be an exciting and practical design problem for engineering. Many difficulties in airfoil design involve the generation of lift, drag, and moment by airfoils sections. Different airfoils have different aerodynamic characteristics that could affect flight performance and safety. Therefore, the calculation of aerodynamic coefficients is a must. These aerodynamic coefficients have received much attention in experimental through a wind tunnel test, theoretical, and numerical studies.

Wind tunnel testing can deliver aerodynamic coefficient results. However, it could take a long time and is more expensive to study airfoil aerodynamics only [1]. Due to recent technology in numerical computation, engineers use various methods to compute aerodynamic coefficients as the first step in airfoil design [2]; then, wind tunnel testing can validate the result of the aerodynamic coefficient.

He et al. use a numerical approach for airfoil design [2]. Another strategy is using Particle Swarm Optimization (PSO) to produce an optimal airfoil design [3]. A spotted hyena optimizer, a metaheuristic approach, is used to optimize airfoil design by minimizing the drag force [4]. The metaheuristic approach belongs to the approximation method, so obtaining an optimal global solution is not guaranteed. Besides that, the metaheuristic has a dilemma in balancing exploration and exploitation. Using data-driven can improve the metaheuristic approach for optimization based on this article [5] to solve that situation.

Nowadays, data size grows unprecedented [6], so airfoil design and analysis based on data-driven is preferable [7]; moreover, the rise of computer science and Artificial Intelligence (AI) has led to the reliable development of the data-driven model. *Yonekura, dkk* [8] dan *Thuerey, dkk* [9] utilize the *Autoencoder* method to explore airfoil design. *Sekar, dkk* [10] use the inverse technique to generate airfoil geometry data based on pressure distribution around the airfoil surface using a Convolutional Neural Network (CNN).

Multiple studies have been applied to predict the aerodynamic coefficient of the airfoil using CNN. This method is known as deep learning and utilizes convolution layers to gather specific patterns from images and use them for training and testing. *Zhang, dkk* [11] use CNN to predict lift force coefficient using an airfoil geometry image as input to the CNN model. On the other side, *Chen, dkk* [12] use CNN to get the lift, drag, and moment predictions. *Hui et al.* [13] utilize CNN to predict the pressure distribution around the airfoil surface. All previous studies need airfoil geometry representations as an input in CNN.

There are several methods to generate airfoil geometry representations. Different airfoil geometry representation methods use various types of parameters. In these articles [11, 12], the grayscale image both outline and solid color of the airfoil represent the airfoil geometry. The grayscale image only has one channel, which has less pixel information

and affects image recognition. To enrich the pixel information, the Signed Distance Function (SDF) [14] uses three channels based on these articles [13, 15].

Nevertheless, airfoil geometry representation using SDF needs additional memory to process the pixel information. Therefore, we propose a new airfoil geometry representation method based on a meshing algorithm to establish the relationship between the airfoil geometry and the CNN architecture. We enrich the pixel information on the grayscale image using a meshing algorithm as the airfoil geometry representation to keep pixel information fast and straightforward pattern recognition on the image.

RESEARCH METHODOLOGY

In general, the steps taken in building a predictive model of the aerodynamic coefficient of the airfoil are, as illustrated in Fig. 1, include airfoil data processing, data selection, and making the trained model.

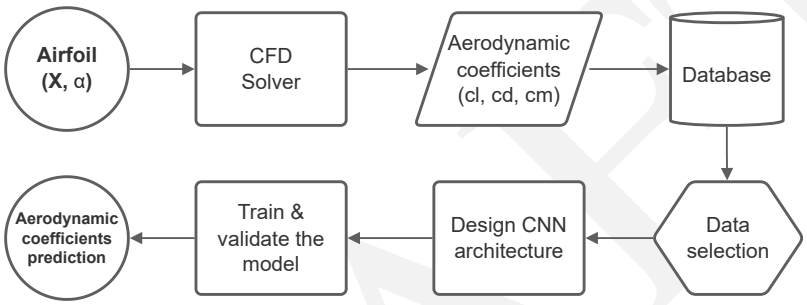


FIGURE 1: Predictive model development flowchart

Airfoil Data Processing

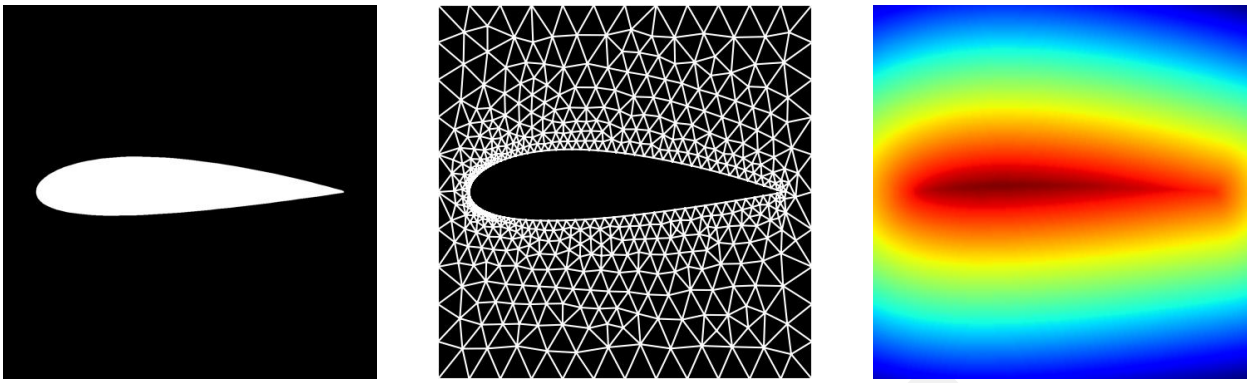
The first step in processing airfoil data in building a predictive model is to obtain the shape or geometry of the airfoil. The airfoil shape consists of coordinates along the top and bottom surfaces. The coordinates are obtained from UIUC Airfoil Data Site [16] and converted into a digital image so that it is easy to implement the architecture with CNN [17]. Airfoil images can be created in the form of a single [11, 12, 18] or three [13, 15] channels using *Signed Distance Field* [19]. This paper uses the *Unstructured Mesh* method [20] to create the airfoil image using the `Gmsh` algorithm [21]. Below is the basic pseudocode to generate airfoil images using the `Gmsh`.

Algorithm 1 Generate airfoil images using mesh

```
procedure MESH
    for point in points do
        hole ← gmsh.add_polygon(point)
        gmsh.add_circle(mean(point), 1.0, holes = [hole])
        mesh ← gmsh.generate_mesh()
        x ← mesh.points[:, 0]
        y ← mesh.points[:, 1]

        plot(x, y)                          ▷ Render the airfoil with mesh
```

Airfoil geometry representations with SDF (Fig. 2c) are more effective with CNN but have more complex information than grayscale (Fig. 2a). Using mesh representation can enrich the pixel information from the grayscale. The mesh images and the airfoil aerodynamic coefficients are stored in the database.



(a) Grayscale image (b) Mesh image (c) SDF image

FIGURE 2: Various airfoil geometry representations

TABLE I: The output resolutions from the last conv layer into fully connected layer

Layer	Aerofoil2BN2FC	Aerofoil3BN2FC	Aerofoil4BN2FC
CONV1	$32 \times 28 \times 28$	$16 \times 57 \times 57$	$10 \times 58 \times 58$
CONV2	$64 \times 6 \times 6$	$32 \times 24 \times 24$	$20 \times 24 \times 24$
CONV3	-	$64 \times 6 \times 6$	$40 \times 10 \times 10$
CONV4	-	-	$60 \times 4 \times 4$

We use `XFOIL` [22] to obtain the airfoil aerodynamic coefficients. `XFOIL` can predict airfoil aerodynamics fastly for a low *Reynold* number [23] using the *potential flow* and *integral boundary layer* method. For flow conditions, we keep $M_a = 0$ and $R_e = 500000$ to get aerodynamic coefficients (C_l, C_d dan C_m) with varying angles of attacks from -20° to 20°. Finally, we keep the airfoil surface roughness configuration N_{crit} = 5 [24].

CNN Architecture

CNN has been widely used in image recognition and was first proposed for gradient-based learning in document recognition [25]. CNN is an ideal architecture to be implemented with image data [26]. CNN uses images with either one channel or three channels as input data. Then the data is transferred to the convolution, pooling and fully connected layers.

As illustrated in Fig. 3, we use LeNet-5 [27] as CNN architecture with some adjustments. The airfoil image has a size of 128×128 as the initial input data. The two-dimensional convolutional (Conv) layer is the main component of CNN. The Conv layer 1 has a kernel size of 15×15 as such in Fig. 3b or 3c, three channels as input and 1 step size in each direction. We use a batch norm to make it faster and more stable during training. The activation function used is a rectified linear unit (ReLU) because it gives better results [28].

In some cases, ReLU activation may experience a Vanishing Gradient during training. Therefore, another activation used is Leaky ReLU [29] as a comparison. After the ReLU operation in Fig. 3c, we set ten channels with a resolution of 58×58 after MaxPool operation based on Eq. 1 in article [30] for the Conv layer 2. We can get output resolution (o) with i as input, p as padding, k as kernel, and s as stride (step) size.

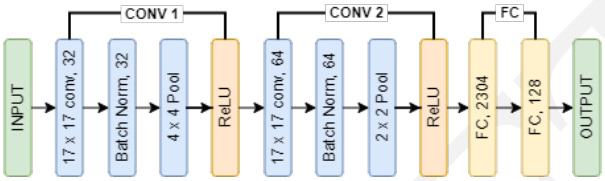
$$o = \left[\frac{i + 2p - k}{s}\right] + 1 \tag{1}$$

We apply the same procedure for the other Conv layers. The last Conv layer will have $60 \times 4 \times 4$ for the first fully connected (FC) layer. The rest of all output resolutions are shown in Table I. The output layer of this architecture is

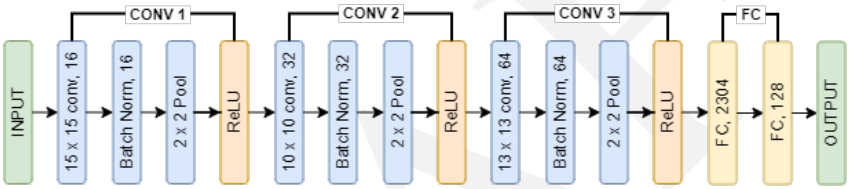
linear regression and uses MSE (Mean Square Error) as a loss function to predict the aerodynamic coefficients of the airfoil.

$$MSE = \sum_{j=1}^{s} \frac{\left(\sum_{i=1}^{n} \frac{(y_i - x_i)^2}{n} \right)_j}{s} \tag{2}$$

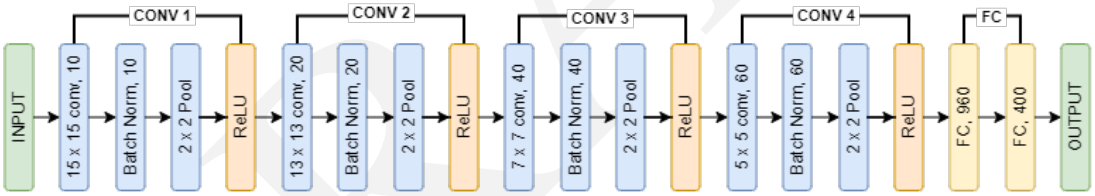
where x_i and y_i are the i^{th} actual and predicted aerodynamic coefficient, respectively; n is the number of airfoil aerodynamic coefficients to be predicted (in this case three coefficients); and s is the batch size.



(a) Aerofoil2BN2FC architecture



(b) Aerofoil3BN2FC architecture



(c) Aerofoil4BN2FC architecture

FIGURE 3: LeNet-5 architecture with modification

Model Training

The model training includes two steps: forward and backward calculation. The first step extracts image features using the convolutional and pooling operations, feeds the features into the fully connected layer, and gets aerodynamic coefficient prediction values through the output layer. Prediction errors are the differences between prediction and actual values. In the last step, the algorithms send these errors back, such as gradient descent, and update the network weights and biases. The number of convolutional, pooling and fully connected layers are flexible depending on the input images.

The forward and backward calculation steps are repeated if the end condition is not reached. The maximal number of epochs determines the end condition of CNN training. The training process of the CNN prediction model is shown in Figure 4.

In this paper, the training algorithm uses adaptive moment estimation (Adam) [31]. Adam minimizes the loss function using a minibatch, a subset of the training set. An epoch is the complete pass of the training algorithm over the entire training set.

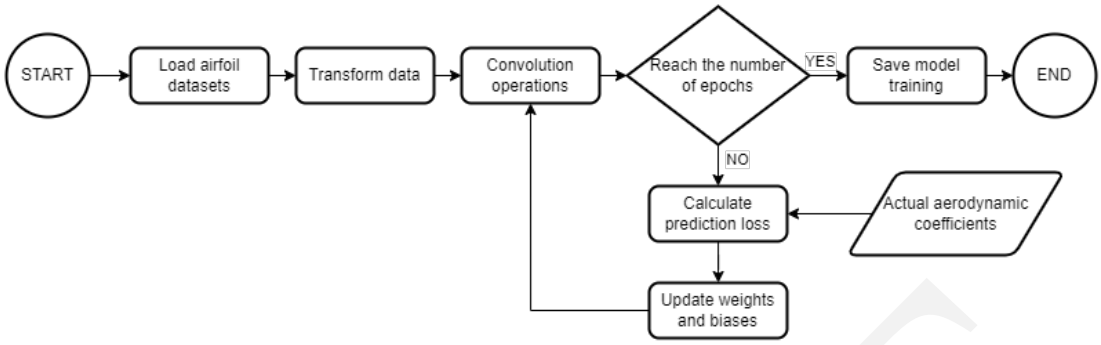


FIGURE 4: Model training procedure

RESULTS AND DISCUSSION

We collect 223 NACA4 airfoils and have 9143 in total datasets. The datasets include airfoil images as input and airfoil coefficients as labels and are split into 80% for training and 20% for validation. Both of these are applied for data transformation. All airfoil images are resized into 128×128 pixels, and then images and airfoil coefficients are standardized using a mean (μ) of 0.5 and standard deviation (σ) of 0.5.

Figure 5 shows the training history for various airfoil geometry representations using the same `Aerofoil4BN2FC` architecture as shown in Figure 3c. The MSE curves for the training datasets drop faster. On the other hand, the MSE curves for validation vary. The validation loss for unstructured mesh does not fall very well and continues flat with the highest MSE loss than others. The grayscale images have lower MSE for validation loss than the unstructured mesh and continue flat for the rest of the epochs. In this training, airfoil images using SDF have the lowest MSE and keep minimizing the error.

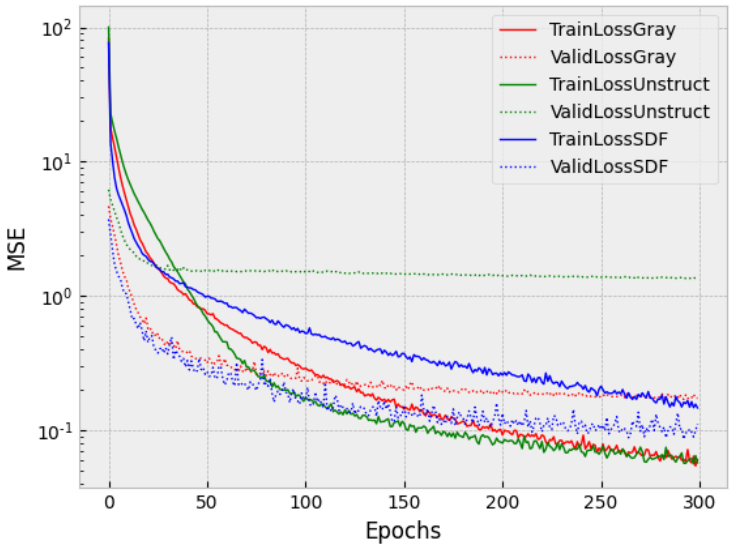


FIGURE 5: Training history for various airfoil geometry representations

Because the SDF can perform very well during training, we increase the maximum epoch number to 1000 and combine it with various architectures as shown in Figure 3. There is no significant improvement when varying the number of convolution layers while validating the datasets, although all architectures show decreasing MSE during training, as shown in Figure 6.

Then, the trained model predicts the aerodynamic coefficients for the unseen airfoil during training. For this purpose, the NACA 2024 has the actual coefficient of lift, drag, and moment from CFD and then compares them to the

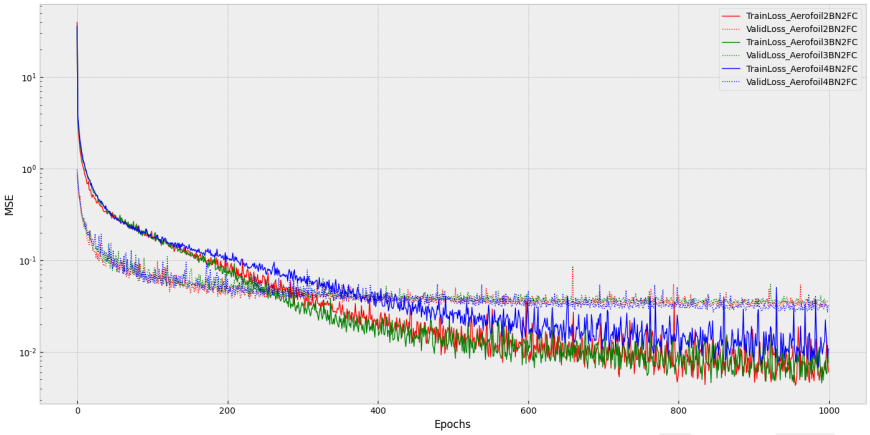
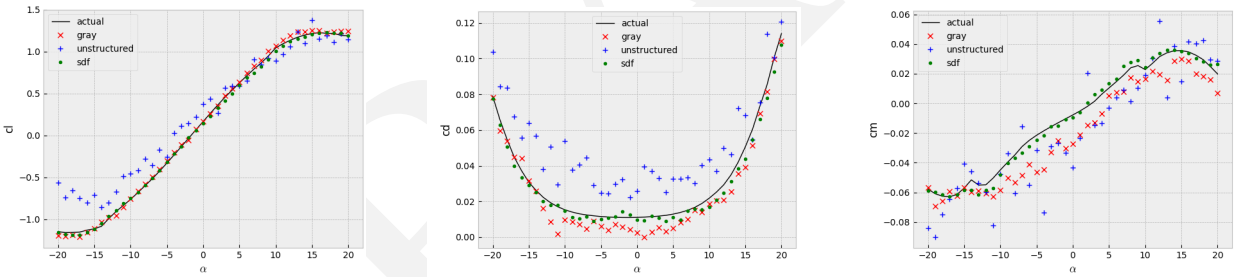


FIGURE 6: Training history for more epochs on SDF datasets

prediction results. From Figure 7a, NACA 2024 image with SDF has the best prediction result. The grayscale image of NACA 2024 has a better prediction result than unstructured mesh. The unstructured mesh cannot perform well at less than zero angle of attack. For the drag coefficient case in Figure 7b, the unstructured mesh performs worst for all angles of attack. The grayscale image has good performance, but the SDF image still has the best prediction results. It also happens on the moment coefficient as shown in Figure 7c, and the SDF image can follow the actual data very well. On the other hand, the grayscale image performs slightly well, and the unstructured mesh still cannot obtain good prediction results.



(a) Lift coefficient (b) Drag coefficient (c) Moment coefficient

FIGURE 7: Aerodynamic coefficient prediction from various geometry representations

The SDF image can perform very well for aerodynamic coefficient. This performance can happen because the trained model using SDF as airfoil geometry representation has the smallest MSE during validation. In this work, Root Mean Squared Error (RMSE) and R-squared (r^2) are metrics to measure the network's performance. Based on Table II, it looks evident that airfoil geometry representation using SDF has the best metric results.

TABLE II: The metric results from various geometry representations using `Aerofoil4BN2FC` architecture

Geometry Representation	*RMSE* C_l	C_d	C_m	r^2 C_l	C_d	C_m
Grayscale	0.0336	0.0067	0.0126	0.9994	0.9822	0.9629
Unstructured Mesh (UM)	0.2337	0.0214	0.0192	0.9848	0.8828	0.8227
SDF	**0.0245**	**0.0036**	**0.0032**	**0.9994**	**0.9952**	**0.9946**
Composite Grayscale [12]	0.0273	0.0035	0.0027	-	-	-
SDF [15]	0.0296	0.0042	0.0042	-	-	-

In terms of the r^2 metric as illustrated in Figure 8, the grayscale image and SDF have a high score in predicting C_l but the UM has slightly below that. The UM is still good for predicting C_l because it can follow the curve pattern in Figure 7a, but it has high value of RMSE, so it cannot perform very well for angles less than zero. Although all geometry representations can follow the curve pattern for predicting C_d, they have good r^2 globally, but only SDF can fit close enough to actual C_d as shown Figure 7b. The grayscale has lower RMSE than the UM. The grayscale and UM have higher RMSE than the SDF for predicting C_m.

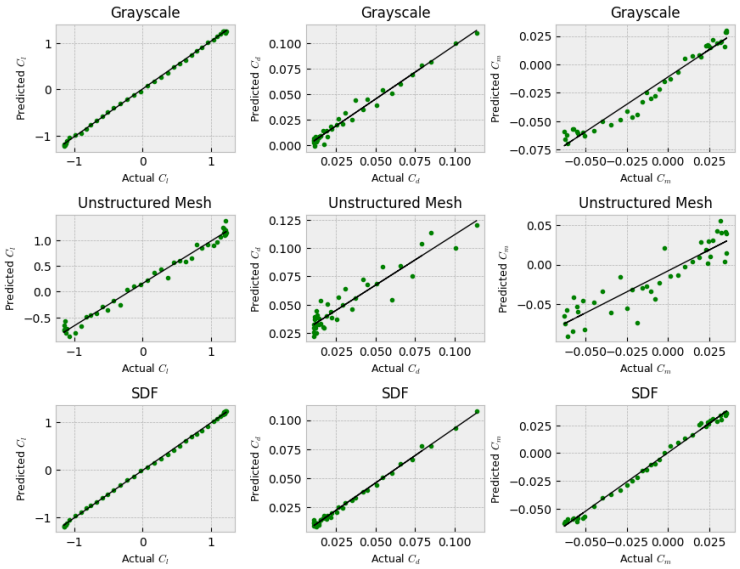


FIGURE 8: The r^2 metric for various prediction result

CONCLUSIONS

This paper proposes a new airfoil geometry representation using a meshing algorithm that makes it easy to train the model in CNN. The prediction model could predict three aerodynamic coefficients of airfoils simultaneously, with little time consumption and high accuracy.

Although there were a few points with substantial deviations from the optimum, most predicted values were close to the actual value, proving the prediction method's feasibility. In future research, we will continue to improve the prediction model.

ACKNOWLEDGEMENT

This work was supported by the National Laboratory for Aerodynamics, Aeroelastics, and Aeroacoustics Technology, National Research and Innovation Agency. The authors would like to thank Hanni Defianti for her help with this project.

REFERENCES

1. J. Tu, G. H. Yeoh, and C. Liu, *Computational fluid dynamics: a practical approach* (Butterworth-Heinemann, 2018).
2. X. He, J. Li, C. A. Mader, A. Yildirim, and J. R. Martins, "Robust aerodynamic shape optimization—from a circle to an airfoil," Aerospace Science and Technology **87**, 48–61 (2019).
3. S. Mirjalili, J. S. Dong, A. Lewis, and A. S. Sadiq, "Particle swarm optimization: theory, literature review, and application in airfoil design," Nature-inspired optimizers , 167–184 (2020).
4. G. Dhiman and A. Kaur, "Optimizing the design of airfoil and optical buffer problems using spotted hyena optimizer," Designs **2**, 28 (2018).

5. B. Crawford, R. Soto, J. Lemus-Romani, M. Becerra-Rozas, J. M. Lanza-Gutiérrez, N. Caballé, M. Castillo, D. Tapia, F. Cisternas-Caneo, J. García, *et al.*, "Q-learnheuristics: towards data-driven balanced metaheuristics," Mathematics **9**, 1839 (2021).
6. Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," Information Fusion **42**, 146–157 (2018).
7. E. Andrés-Pérez, "Data mining and machine learning techniques for aerodynamic databases: Introduction, methodology and potential benefits," Energies **13**, 5807 (2020).
8. K. Yonekura and K. Suzuki, "Data-driven design exploration method using conditional variational autoencoder for airfoil design," Structural and Multidisciplinary Optimization **64**, 613–624 (2021).
9. N. Thuerey, K. Weißenow, L. Prantl, and X. Hu, "Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows," AIAA Journal **58**, 25–36 (2020).
10. V. Sekar, M. Zhang, C. Shu, and B. C. Khoo, "Inverse design of airfoil using a deep convolutional neural network," Aiaa Journal **57**, 993–1003 (2019).
11. Y. Zhang, W. J. Sung, and D. N. Mavris, "Application of convolutional neural network to predict airfoil lift coefficient," in *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* (2018) p. 1903.
12. H. Chen, L. He, W. Qian, and S. Wang, "Multiple aerodynamic coefficient prediction of airfoils using a convolutional neural network," Symmetry **12**, 544 (2020).
13. X. Hui, J. Bai, H. Wang, and Y. Zhang, "Fast pressure distribution prediction of airfoils using deep learning," Aerospace Science and Technology **105**, 105949 (2020).
14. S. Osher and R. Fedkiw, "Signed distance functions," in *Level set methods and dynamic implicit surfaces* (Springer, 2003) pp. 17–22.
15. Z. Yuan, Y. Wang, Y. Qiu, J. Bai, and G. Chen, "Aerodynamic coefficient prediction of airfoils with convolutional neural network," in *Asia-Pacific International Symposium on Aerospace Technology* (Springer, 2018) pp. 34–46.
16. M. S. Selig, "Uiuc airfoil data site," (1996).
17. S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)* (Ieee, 2017) pp. 1–6.
18. J. Viquerat and E. Hachem, "A supervised neural network for drag prediction of arbitrary 2d shapes in laminar flows at low reynolds number," Computers & Fluids **210**, 104645 (2020).
19. S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik, "Prediction of aerodynamic flow fields using convolutional neural networks," Computational Mechanics **64**, 525–545 (2019).
20. D. Supriya, K. Nagaraja, T. Smitha, and S. Jayan, "Accurate higher order automated unstructured triangular meshes for airfoil designs in aerospace applications using parabolic arcs," Aerospace Science and Technology **88**, 405–420 (2019).
21. C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities," International journal for numerical methods in engineering **79**, 1309–1331 (2009).
22. M. Drela, "Xfoil: An analysis and design system for low reynolds number airfoils," in *Low Reynolds number aerodynamics* (Springer, 1989) pp. 1–12.
23. J. Morgado, R. Vizinho, M. Silvestre, and J. Páscoa, "Xfoil vs cfd performance predictions for high lift low reynolds number airfoils," Aerospace Science and Technology **52**, 207–214 (2016).
24. H. Y. Mark Drela, "Xfoil 6.9 user primer,".
25. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE **86**, 2278–2324 (1998).
26. I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning* (MIT press, 2016).
27. Y. LeCun *et al.*, "Lenet-5, convolutional neural networks," URL: http://yann. lecun. com/exdb/lenet **20**, 14 (2015).
28. X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (JMLR Workshop and Conference Proceedings, 2011) pp. 315–323.
29. A. K. Dubey and V. Jain, "Comparative study of convolution neural network's relu and leaky-relu activation functions," in *Applications of Computing, Automation and Wireless Systems in Electrical Engineering* (Springer, 2019) pp. 873–880.
30. V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," arXiv preprint arXiv:1603.07285 (2016).
31. K. Bae, H. Ryu, and H. Shin, "Does adam optimizer keep close to the optimal point?" arXiv preprint arXiv:1911.00289 (2019).