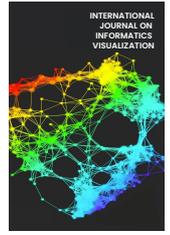




# INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : [www.joiv.org/index.php/joiv](http://www.joiv.org/index.php/joiv)



## Convolutional Neural Networks-Based for Predicting Aerodynamic Coefficient of Airfoils at Ultra-Low Reynolds Number

Alief Sadlie Kasman <sup>a,c,\*</sup>, Arizal Akbar Zikri <sup>a,c</sup>, Fariduzzaman <sup>c</sup>, Wahyu Srigutomo <sup>b</sup>

<sup>a</sup> Department of Computational Science, Institut Teknologi Bandung, Coblong, Bandung, 40116, Indonesia

<sup>b</sup> Physics Department, Institut Teknologi Bandung, Coblong, Bandung, 40116, Indonesia

<sup>c</sup> National Laboratory for Aerodynamics, Aeroelastic, and Aeroacoustics Technology, National Research and Innovation Agency, Setu, Tangerang Selatan, 15314, Indonesia

Corresponding author: \*alie001@brin.go.id

**Abstract**— Many applications, including airplane design, wind turbines, and heat transmission, use symmetric or asymmetric airfoils. Engineers employ these airfoil shapes to optimize performance and efficiency. Each airfoil has a unique set of aerodynamic coefficients that must be calculated to maximize the airfoil design. Engineers utilize numerous ways to calculate coefficients, such as lift and drag. One of the methods is the prediction method, which effectively reduces time and cost. This study's training dataset is obtained from particle-based numerical computation using the Lattice Boltzmann Method (LBM). Then, Convolutional Neural Networks (CNN) are used as a prediction method to get the aerodynamic coefficients of airfoils for lift and drag based on two different Reynolds numbers. In CNN, airfoil geometry representation is essential. The Signed Distance Function (SDF) was used to convert airfoil geometry into RGB pictures. On the other hand, the SDF method cannot explain different flow conditions; in this case, it is represented by the Reynolds number ( $Re$ ). Therefore, we propose a Text-based Watermarking Method (TWM) to differentiate between  $Re = 500$  and  $Re = 1000$ . Each airfoil representation was trained and tested to generate each prediction model using a modified LeNet-5. The computation results show that using CNN with TWM on SDF to define the Reynolds numbers could predict the lift and drag coefficients with varying angles of attack. Future research can focus on generalizations to different aerodynamic aspects and practical applications in complex scenarios.

**Keywords**— Neural network; CNN; airfoil; aerodynamic coefficient; LBM.

Manuscript received 5 Oct. 2023; revised 13 Dec. 2023; accepted 7 Jan. 2024. Date of publication 31 Mar. 2024.  
International Journal on Informatics Visualization is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



### I. INTRODUCTION

The human desire to fly has been around for a long time. The phenomenon of natural flight by birds or bees inspires this desire [1]–[3]. These articles explain the aspects needed to mimic that flight motion, but there is not much explanation related to aerodynamic analysis. This analysis requires conditions at ultra-low Reynolds ( $Re$ ) numbers [4], thus making this study attract a lot of attention among researchers [5]. In recent years, there has been notable interest in ultra-low Reynolds number aerodynamics, primarily driven by the increased prevalence of micro-aerial vehicles operating at low speeds.

Using an experimental or numerical computation approach, we can gain knowledge of aerodynamic characteristics at ultra-low  $Re$ . From the testing, the analysis is more difficult when using air media, so the medium used is water. Although

this approach is possible, it becomes difficult to apply in an experimental setup if  $Re$  is much lower than 7000 [6].

For this reason, many researchers use a numerical computational approach in flow analysis at ultra-low  $Re$ . Papers [7]–[9] compute flow simulation intensively for airfoil NACA 0012 at  $Re = 1000$  by varying the angle of attack. Besides, Suzuki et al. [10] simulate wing flapping using the lattice Boltzmann method (LBM) at  $Re = 100$ . Many applications can be applied through this LBM [11], including the fluid flow around the airfoil at ultra-low  $Re$  and the calculation of aerodynamic coefficients such as the coefficients of lift ( $C_l$ ) and drag ( $C_d$ ) [12]–[14]. The attractiveness of LBM in simulating fluid flows lies in the benefits of numerical computation compared to other CFD methods, such as easy-to-follow algorithms, efficient implementation of parallel computations, handling of complex geometries, and so on [15].

Apart from the advantages of LBM, not all experts can apply LBM to obtain aerodynamic coefficient datasets. Then, the idea emerges to create a model that can predict the value of  $C_l$  and  $C_d$  when a user gives information about airfoil geometry, angle of attack, and  $Re$ , so that the user does not have to deal directly with the LBM. A data-driven modeling(DDM) approach is needed to support the idea. DDM is a data analysis process that finds the relationship between input (airfoil geometry, angle of attack, and  $Re$ ) and output ( $C_l$  and  $C_d$ ) [16]. An article [17] utilizes DDM to obtain the required airfoil geometry through the conditional variational autoencoder (CVAE). Another DDM approach in [18] is predicting the airfoil surface pressure distribution through a convolutional neural network (CNN). CNN requires images as input data [19] and has the advantage of automatically recognising patterns from images without human supervision [20]. CNN has many architectures that can map between image features with regressions or classifications. One of the well-known CNN architectures in digital recognition is LeNet-5 [21]. LeNet-5 requires few parameters but performs well during model training [22].

LeNet-5 was initially used for classification, but this architecture can be modified to predict aerodynamic coefficients. In article [23],  $C_l$ ,  $C_d$ , and pitch moment ( $C_m$ ) were obtained using XFOIL. The airfoil geometries were converted into images through the Signed Distance Field (SDF) method, and then the modified LeNet-5 was used to predict the aerodynamic coefficient. LeNet-5 was also used to predict  $C_l$  in [24]. The  $C_l$  datasets were also obtained using XFOIL, while the airfoil geometries were converted into grayscale images. Mach ( $Ma$ ) and  $Re$  number were injected in the first fully connected (FC) layer. Although not only for airfoils, article [25] utilized the modified LeNet-5 to predict  $C_d$  with arbitrary geometries, including airfoils. The geometries were converted into grayscale images. The  $C_d$  datasets were obtained using the RANS method.  $C_l$ ,  $C_d$ , and  $C_m$  were also predicted in [26] using the modified LeNet-5 and composite technique to generate airfoil images.

This article uses LBM to collect  $C_l$  and  $C_d$  datasets for conditions  $Re = 500$  and  $Re = 1000$ .  $Re = 1000$  is primarily used to validate the results of fluid flow simulations with LBM and can also be compared with the findings of previous studies.  $Re = 500$  is employed to test the capabilities of the CNN model trained under different Reynolds number conditions. The SDF method converts the airfoil geometries into images [27]. We introduce the Text-based Watermarking Method on SDF (TWM+SDF), a new and fast technique to define different flow conditions. This technique does not require complex calculations, making it expected to be easily followed by other researchers. Next, we modify LeNet-5 by varying the depth of the convolution layers, the activation function between ReLU and Leaky ReLU, and the Adam optimizer's learning rate (LR).

## II. MATERIALS AND METHOD

As shown in Fig. 1, the procedures involved in developing a predictive model of the aerodynamic coefficient of an airfoil include airfoil processing, data transformation, and the creation of the trained model.

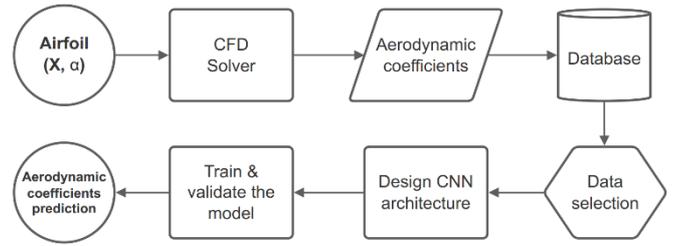
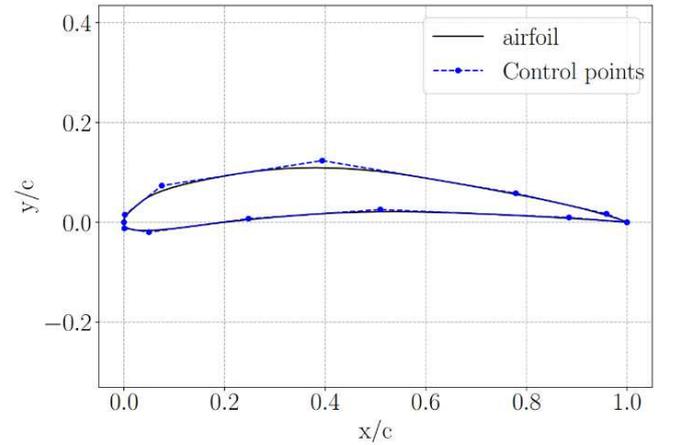


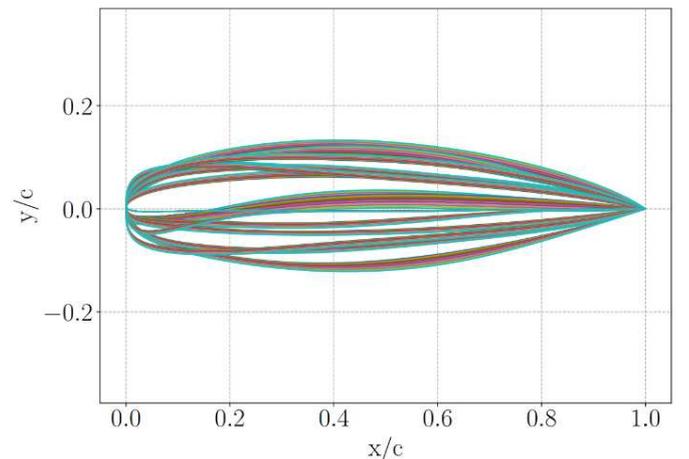
Fig. 1 Predictive model development flowchart

### A. Airfoil Data Processing

The first stage in processing airfoil data for predictive modeling is determining the form or geometry of the airfoil. Coordinates define the top and bottom surfaces of an airfoil. The airfoil coordinates are generated using Non-Uniform Rational B-Splines (NURBS) [28] with 12 control points, as shown in Fig. 2a. We develop 110 airfoils by linearly varying control points at the upper and lower surfaces, as shown in Fig. 2b. Then, all airfoils are converted into digital images, making it easy to implement with CNN [29].



(a) Airfoil NURBS with 12 control points



(b) Linear variations on control points

Fig. 2 The process of producing various airfoils

Airfoil images can be created in the form of single [24]–[26] or three [18], [23] channels using a Signed Distance Field (SDF) [30]. This paper uses the SDF method to create airfoil images. We use TWM on SDF to encounter different Reynolds numbers, as illustrated in Fig. 3. The database stores

the SDF pictures as well as the aerodynamic coefficients of the airfoil.

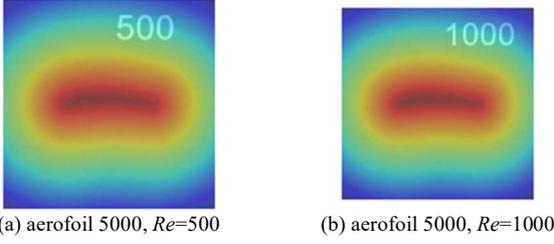


Fig. 3 Text-based watermarking methods on SDF

### B. Lattice Boltzmann Method

As our CFD solver, the LBM has many great features in fluid simulation among the many numerical methods in the CFD fields. For ultra-low Reynolds numbers, the aerodynamic coefficient of airfoils can be computed using LBM [12]. At a microscopic level, the particles of a fluid live on a lattice. This paper uses a two-dimensional lattice with nine possible velocities at each lattice (D2Q9), as described in Fig. 4.

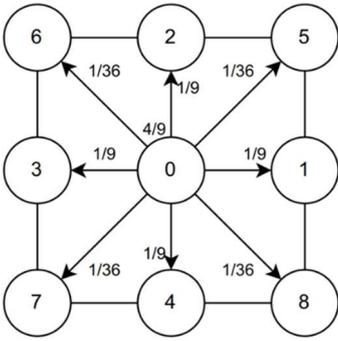


Fig. 4 The two-dimensional scheme for particles in the lattice

The microscopic particles that build up fluid can be explained by the distribution function  $f_i$ , which describes the phase-space density at a particular location and velocity. The particles will undergo two processes: stream and collide.

The BGK [31] approximation can capture that behavior using the discretized equation below :

$$f_i(x_i + v_i \Delta t, t + \Delta t) - f_i(x_i, t) = -\frac{f_i(x_i, t) - f_i^{eq}(x_i, t)}{\tau} \quad (1)$$

where  $i$  denotes nine lattice directions,  $\tau$  is the timescale of collisions, and the distribution function  $f$  tends towards some equilibrium state  $f^{eq}$ .

$$f_i^{eq} = w_i \rho (1 + 3(v_i \times u) + \frac{9}{2}(v_i \times u)^2 + \frac{3}{2}(u \times u)^2) \quad (2)$$

ferocity in the lattice, and  $w_i = \begin{cases} \frac{4}{9}, & \text{for } i = 0 \\ \frac{1}{9}, & \text{for } i = 1, 2, 3, 4 \\ \frac{1}{36}, & \text{for } i = 5, 6, 7, 8 \end{cases}$

$$v_i = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \end{bmatrix}$$

The calculation of the macro variables, such as density and velocity, can be obtained using the following equation:

$$\begin{aligned} \rho &= \sum f_i \\ \rho u &= \sum f_i v_i \end{aligned} \quad (3)$$

The formula used to calculate the lift and drag coefficient can be shown below:

$$\begin{aligned} C_l &= \frac{F_y}{0.5 \rho c U^2} \\ C_d &= \frac{F_x}{0.5 \rho c U^2} \end{aligned} \quad (4)$$

where  $c$  is the chord of an airfoil,  $U$  is the velocity in lattice units, and

$$F_{x|y} = \sum v_i (f(s, t) + f(w, t)) \quad (5)$$

where  $s$  is the node in the solid and  $w$  is the node of the fluid adjacent to  $s$ .

The calculation process using LBM can be shown in the following Fig. 5. The simulation stop criteria are based on the total number of iterations used.

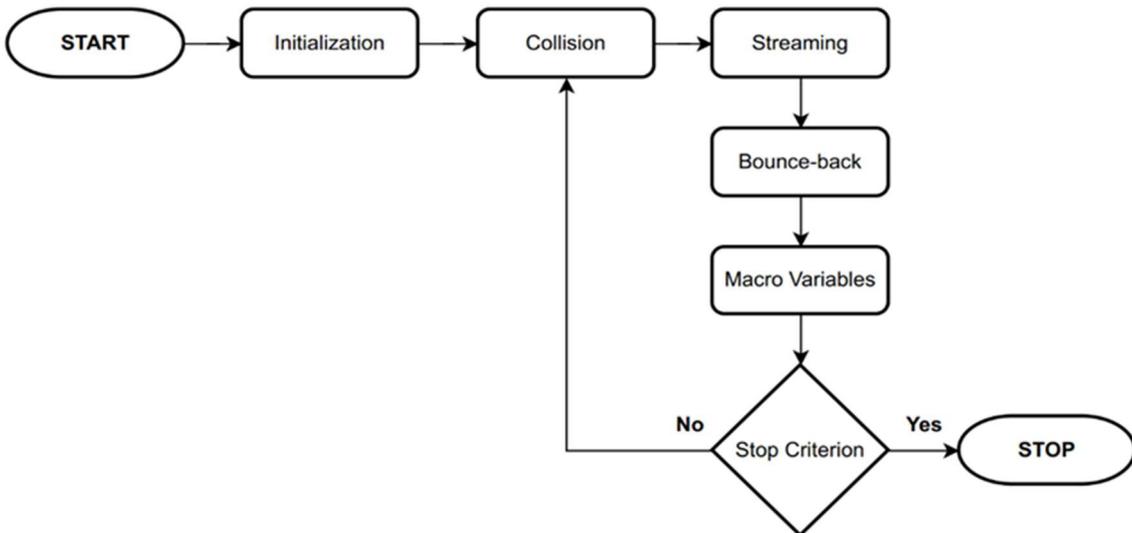


Fig. 5 Calculation process used in LBM

### C. CNN Architecture

CNN has been widely utilized in image recognition and was first introduced in document recognition for gradient-based learning [32]. CNN is an excellent architecture for

implementing image data [33]. As input data, CNN uses pictures with one or three channels. The data is subsequently forwarded to the convolution, pooling, and fully connected layers.

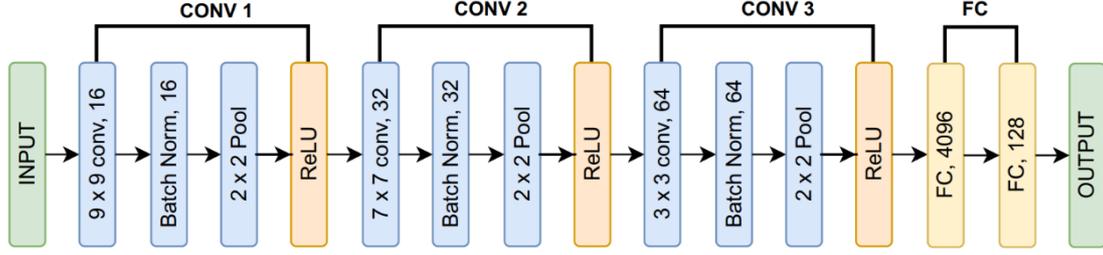


Fig. 6 LeNet-5 architecture with modification

As shown in Fig. 6, we employ LeNet-5 as a CNN architecture with certain modifications. As the first input data, the airfoil image has a size of  $78 \times 78$ . CNN's major component is the two-dimensional convolutional (Conv) layer. Conv layer 1 has a kernel size of  $9 \times 9$ , three input channels, and one step size in each direction. We employ a batch norm to make training faster and more stable. Because it produces better results, the activation function utilized is a rectified linear unit (ReLU) [34].

In some cases, ReLU activation may experience a Vanishing Gradient during training. Therefore, another activation used is Leaky ReLU [35] as a comparison. After the ReLU operation in Fig. 6, we set 32 channels with a resolution of  $16 \times 16$  after the MaxPool process based on Eq. 6 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.

$$o = \left\lfloor \frac{i+2p-k}{s} \right\rfloor + 1 \quad (6)$$

The same process is used for the other Conv layers. For the first fully connected (FC) layer, the last Conv layer will have  $64 \times 8 \times 8$ . This architecture's output layer is linear regression, with MSE (Mean Square Error) in Eq. 7 as a loss function.

$$MSE = \sum_{j=1}^s \frac{\left( \frac{\sum_{i=1}^{num} (\hat{y}_i - y_i)^2}{num} \right)_j}{s} \quad (7)$$

The output layer is used to predict the coefficient of lift and drag. where  $y_i$  and  $\hat{y}_i$  are the  $i^{th}$  LBM and predicted aerodynamic coefficients, respectively;  $num$  is the number of coefficients to be expected; and  $s$  is the group size.

### D. Model Training

The model training procedure is divided into two stages: forward and backward computations. The first phase extracts picture patterns using convolutional and pooling algorithms and passes them into the FC layer. The projected aerodynamic coefficient may then be obtained using the output layer. The disparities between actual and expected values are referred to as prediction errors. Finally, the algorithms feed these mistakes back into the network, adjusting its weights and biases. The amount of Conv, pooling, and FC layers is adjustable based on the input images. If the end condition is not met, the forward and backward computation steps are repeated. The total number of epochs is the stopping condition for CNN training. Fig. 7 depicts the training approach for obtaining the prediction model.

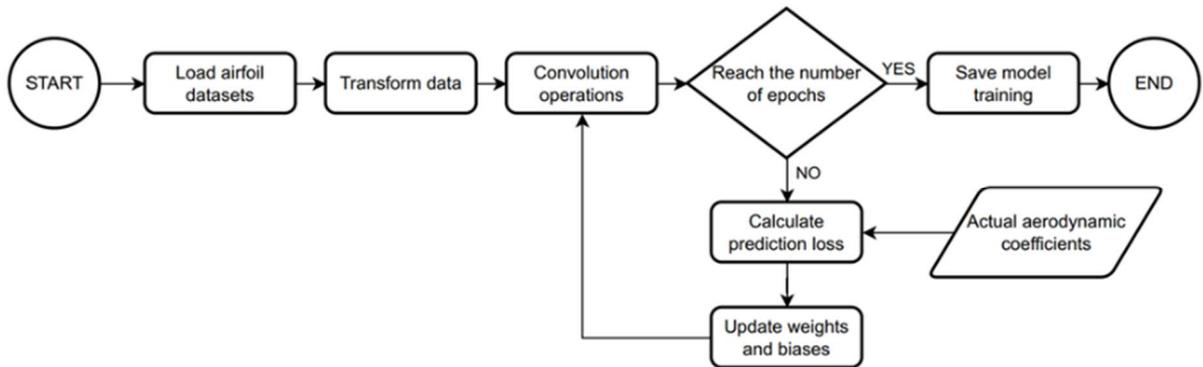


Fig. 7 Model training procedure

The training approach in this study employs adaptive moment estimation (Adam). Adam uses a portion of the training dataset to minimize the loss function. The number of epochs is the amount of time required to run the learning algorithm through the full training dataset.

## III. RESULTS AND DISCUSSION

### A. Simulation Using LBM

The flow simulation process around the airfoil using LBM is carried out for 30000 iterations for each airfoil. The

variation of the angle of attack starts from  $0^\circ$  to  $29^\circ$ . In this study, we use two variations of the Reynolds number:  $Re =$

$500$  and  $Re = 1000$ . The simulation domain can be shown in the following Fig. 8.

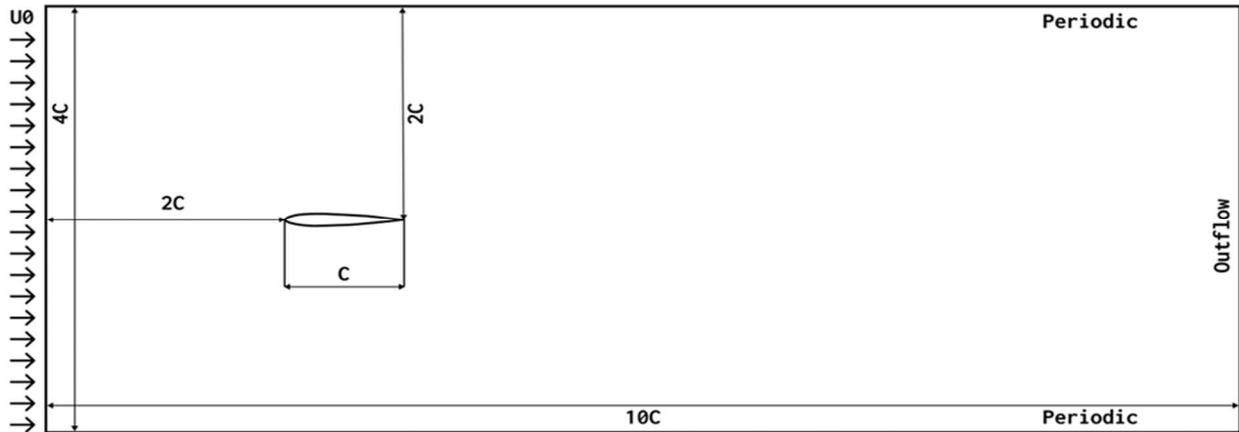


Fig. 8 Simulation domain configuration

Each side of the domain is assigned a boundary condition. In our study, the boundary conditions used for the upper and lower sides are periodic. With this condition, the distribution function ( $f_i$ ) coming out of the top side will be the same as the distribution function ( $f_i$ ) coming in from the bottom. The left side of the domain has a boundary condition in the form of a specified initial velocity profile. In contrast, the right side satisfies the Neumann boundary condition so that the

distribution function  $f_i$  of the right side of the domain is the same as the left side of its neighbors. The bounceback boundary condition is used on the airfoil.

As shown in Fig. 9, the simulation results are carried out at  $Re = 500$  and  $Re = 1000$  for NACA 0012. The variation of the angle of attack is  $30^\circ$ , ranging from  $0^\circ$  to  $29^\circ$ , with the step increment being  $1^\circ$ .

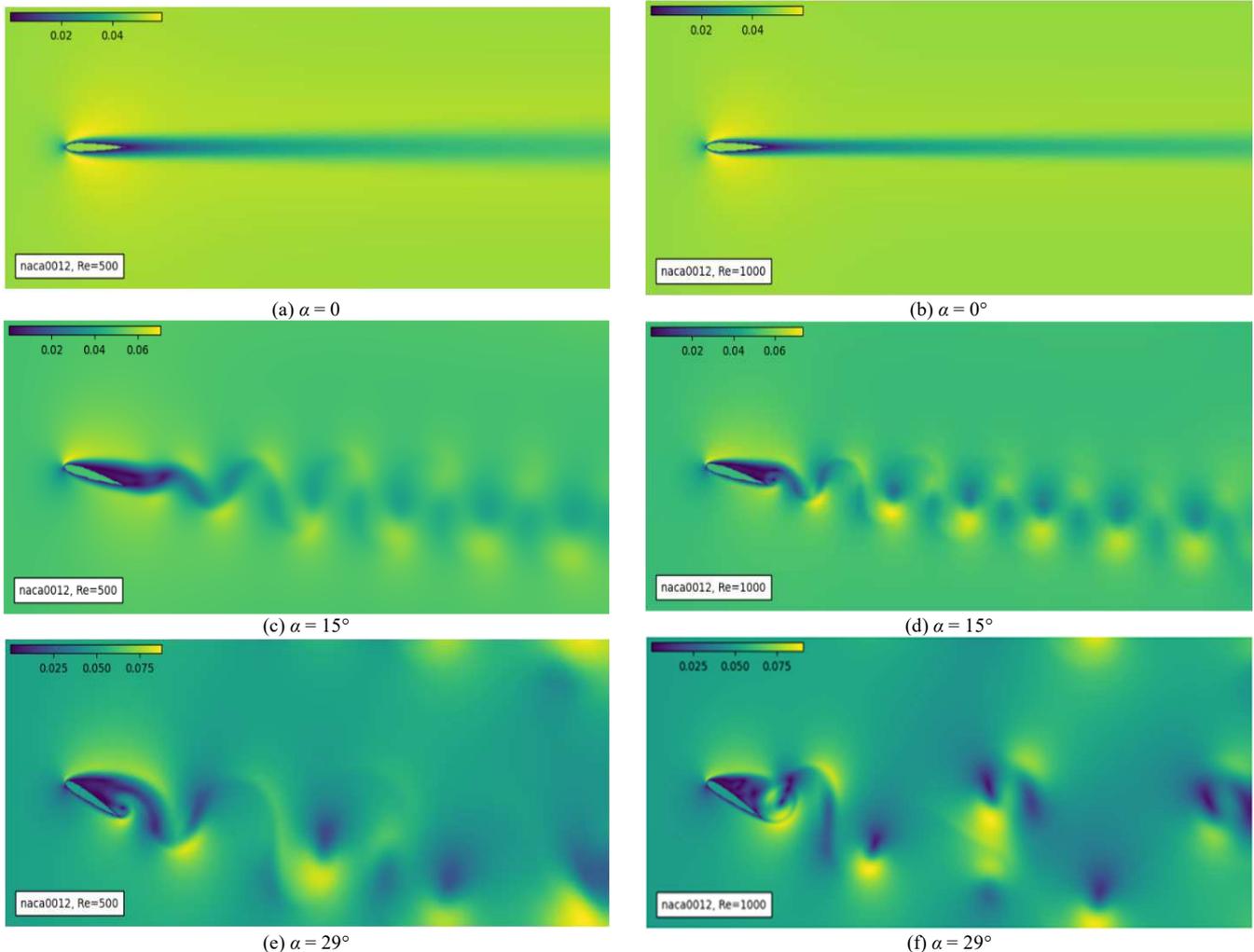
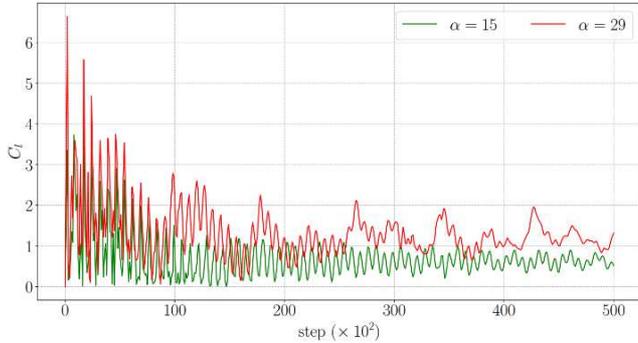
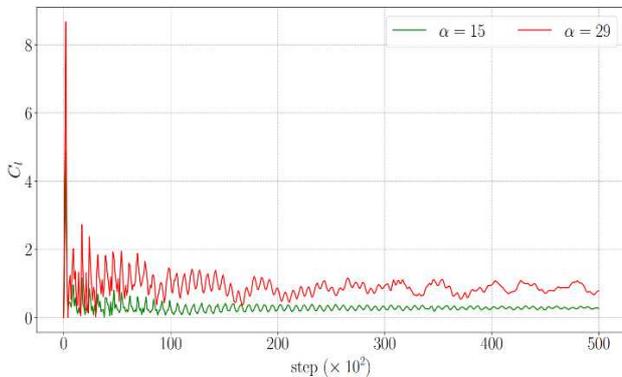


Fig. 9 Flow simulation around the airfoil at  $Re = 500$  and  $Re = 1000$

The  $C_l$  and  $C_d$  are calculated every 100 iterations of 50000 using Eq. 4. In Fig. 10, The time history in the lattice unit of the coefficient lift and drag is compared to different angles of attack during the simulation. For NACA 0012 with  $\alpha = 15^\circ$ ,  $C_l$  is more stable, starting at 10000 iterations. As for  $\alpha = 29^\circ$ ,  $C_l$  fluctuates due to irregular flow behind the airfoil, as shown in Fig. 9f.

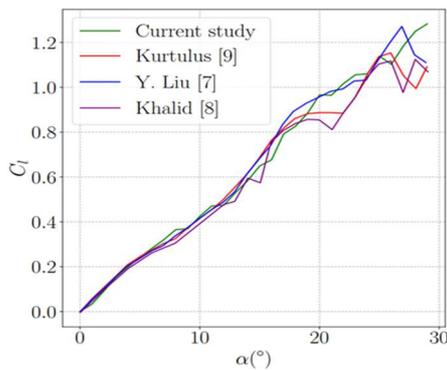


(a) The time history of the coefficient lift

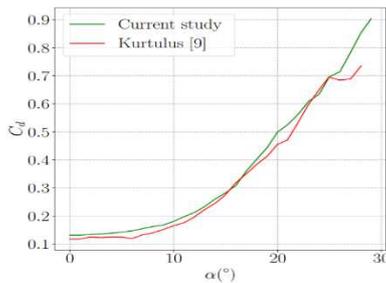


(b) The time history of the coefficient drag

Fig. 10 The comparison of the coefficient lift and drag on different angles of attacks during iteration.



(a)  $C_l$  against  $\alpha$  at  $Re = 1000$



(b)  $C_d$  against  $\alpha$  at  $Re = 1000$

Fig. 11 The comparison of  $C_l$  and  $C_d$  values at  $Re = 1000$

To validate the simulation results, we compared the results of the calculation of the average aerodynamic coefficient in the LBM with the results carried out by Kurtulus et al. [9] and plotted them in Fig. 11. In general, there are similarities between the numerical solutions from Kurtulus et al. and the current study, as shown in Fig. 11a. At about  $\alpha = 26^\circ$ , the value of  $C_l$  has decreased; a separate fluid flow could cause this. At  $\alpha = 0^\circ$  to  $\alpha = 6^\circ$ , the value of  $C_d$  changes minimally.

From  $\alpha = 8^\circ$ , the value of  $C_d$  starts to increase due to the vortex shedding behind the airfoil. The calculation result of  $C_d$  can be plotted and shown in Fig. 11b. It turns out that the results between them are acceptable. Based on these results, simulations with LBM can be applied to 110 airfoils using NURBS. Simulations are performed at  $Re = 500$  and  $Re = 1000$ .

### B. Prediction Using CNN

We gather 110 airfoils and have a total of 6600 datasets. The datasets are separated into 80% for training and 20% for validation and comprise airfoil pictures as input and airfoil coefficients as labels. These are both used to transform data. The geometry representations and airfoil coefficients are then standardized with a mean ( $\mu$ ) of 0.5 and a standard deviation ( $\sigma$ ) of 0.5 for all airfoil pictures.

Fig. 12 shows the training history for ReLU and leaky-ReLU activation functions. The MSE curves for the training datasets drop faster; then, the MSE remains constant around 1000 epochs. On the other hand, the MSE curves for validation decrease more quickly until about 100 epochs. Both ReLU and Leaky-ReLU perform very well in this case. So, airfoil images using SDF have a good result in the training process.

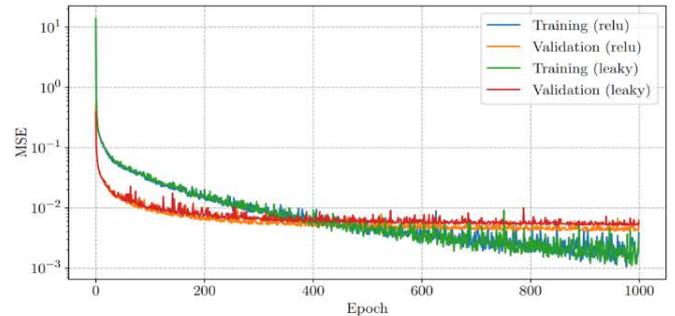
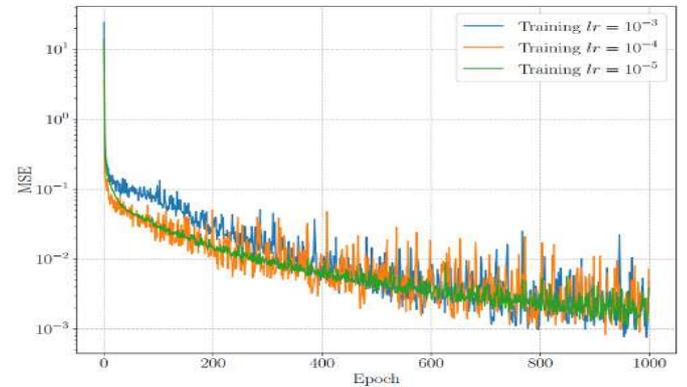
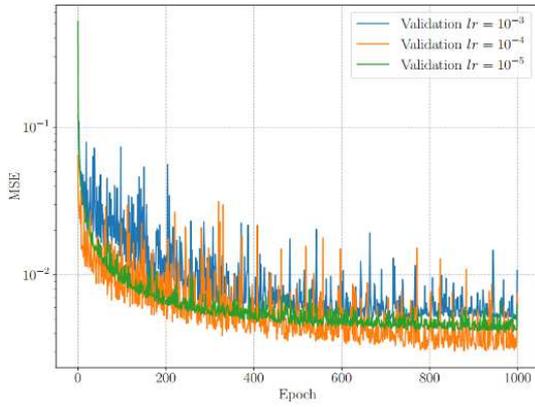


Fig. 12 The training and validation history

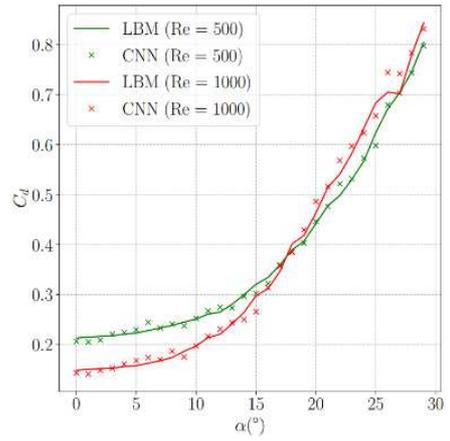


(a) Training history



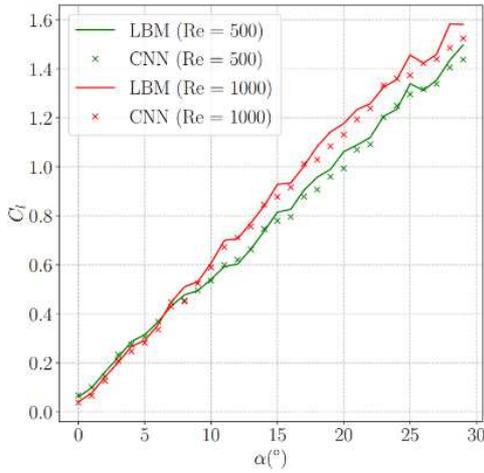
(b) Validation history

Fig. 13 Comparison of training and validation with different learning rates

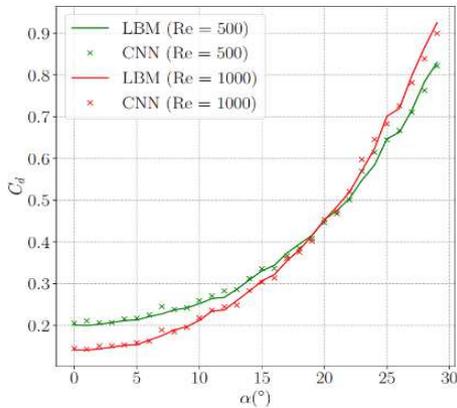


(d)  $C_d$  for NACA 0018

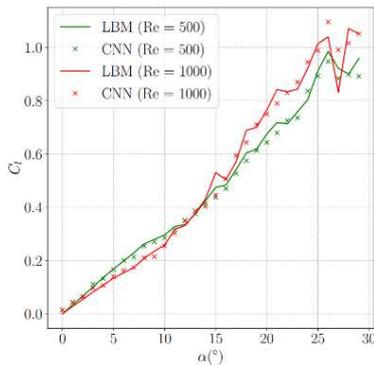
Fig. 14 Aerodynamic coefficient prediction for two different Reynolds number



(a)  $C_l$  for NACA 6412



b)  $C_d$  for NACA 6412



(c)  $C_l$  for NACA 0018

Because the SDF can perform very well during training, we vary the learning rate ( $lr$ ) to  $10^{-3}$ ,  $10^{-4}$ , and  $10^{-5}$  with the same ReLU activation function. Based on Fig. 13, loss fluctuations in the training and validation stages,  $lr = 10^{-5}$ , have the slightest MSE fluctuation among the other  $lr$  values. So, we focus on  $lr = 10^{-5}$  and the ReLU activation function for our training model.

To evaluate model performance, the trained model calculates the aerodynamic coefficients for the unknown airfoil during training. The NACA 6412 compares the actual lift and drag coefficients from the preceding section to the predicted findings. From Fig. 14, the model can perform prediction very well on NACA 6412 and NACA 0018 for  $C_l$  and  $C_d$  both at  $Re = 500$  and  $Re = 1000$ . Differences in prediction results can occur, especially at large angles of attack. These differences may be caused by  $C_l$  and  $C_d$  value fluctuations, as shown in Fig. 10. However, using TWM on SDF, this model still performs very well for different Reynolds numbers.

The CNN model performance using TWM on SDF as an airfoil geometry representation can be quantified using Root Mean Squared Error (RMSE) and R-squared ( $r^2$ ). Both are evaluation metrics for regression purposes, as they measure how close the predicted values are to the actual values, as previously utilized in prior studies [23]–[27].

Based on Table 1, TWM+SDF at  $Re = 500$  has a better metric than  $Re = 1000$  for both  $C_l$  and  $C_d$ . In addition, the RMSE value of  $C_d$  is smaller than that of  $C_l$ . However, this is still within the acceptable range by comparing the general results from ref. [23] and [26].

TABLE I  
THE METRIC MEASUREMENT RESULTS

TWM+SDF	RMSE		$r^2$	
	$C_l$	$C_d$	$C_l$	$C_d$
NACA 6412 ( $Re = 500$ )	0.0265	0.0103	0.9978	0.9973
NACA 6412 ( $Re = 1000$ )	0.0367	0.0118	0.9971	0.9979
NACA 0018 ( $Re = 500$ )	0.0231	0.0095	0.9964	0.9972
NACA 0018 ( $Re = 1000$ )	0.0404	0.0164	0.9867	0.9950

Regarding the  $r^2$  metric, as illustrated in Fig. 15, the TWM+SDF at  $Re = 500$  has a higher score in predicting  $C_l$  than at  $Re = 1000$ , which shows that the CNN model can better understand the variation of the overall prediction data. In

contrast to the  $C_d$ , the CNN model has a higher score for  $Re = 1000$ . However, the CNN model has a high  $r^2$  score when predicting  $C_l$  and  $C_d$  for different Reynolds numbers.

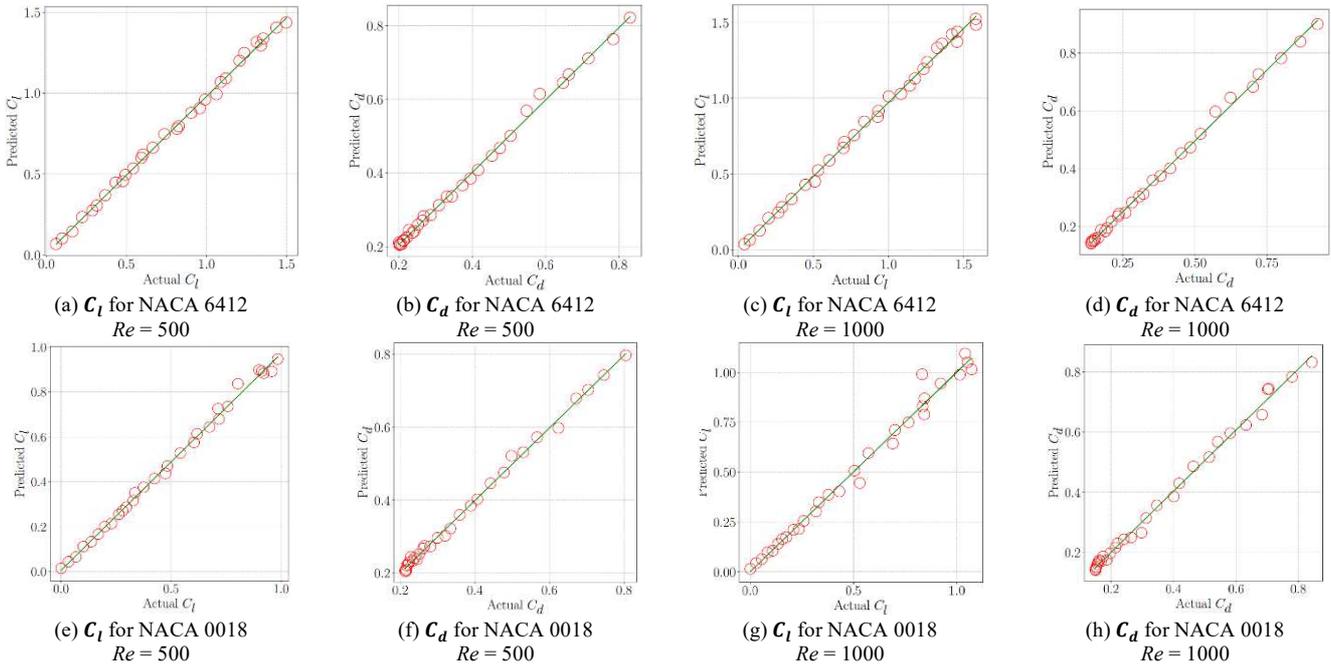


Fig. 15 The  $r^2$  metric for various prediction results

#### IV. CONCLUSION

The previous section utilized LBM to simulate fluid flow around the airfoil at ultra-low Reynolds numbers. The results demonstrated excellent performance compared to previous studies. LBM has the potential to conduct fluid flow simulations due to its simpler computational algorithms, easy and efficient implementation for parallel computing, and easier handling of complex geometries. LBM is open source, allowing for its development for other research purposes.

In addition to the aerodynamic coefficients obtained using LBM for different Reynolds numbers, airfoil geometries with TWM+SDF are also utilized as datasets. These datasets were trained using CNN with the LeNet-5 architecture to predict airfoil lift and drag coefficients. Predictive performance is measured by metrics and using TWM+SDF can yield good prediction results for various Reynolds Numbers. Both RMSE and R-squared metrics provide equally satisfactory results. This positive prediction outcome will create new opportunities for researchers or other communities to collect more training data at different Reynolds numbers for airfoil design. However, it's important to note that this research is currently limited to 2D airfoil cases, and further investigations into more complex scenarios, such as aircraft wings, would be intriguing to pursue.

#### ACKNOWLEDGMENT

This work is supported by the National Laboratory for Aerodynamics, Aeroelastic, and Aeroacoustics Technology (LA3 BRIN). The computation in this work has been done

using the facilities of MAHAMERU BRIN HPC, National Research and Innovation Agency of Indonesia (BRIN).

#### REFERENCES

- [1] H. V. Phan and H. C. Park, "Insect-inspired, tailless, hover-capable flapping-wing robots: Recent progress, challenges, and future directions," *Progress in Aerospace Sciences*, vol. 111, p. 100573, Nov. 2019, doi: 10.1016/j.paerosci.2019.100573.
- [2] N. T. Jafferis, E. F. Helbling, M. Karpelson, and R. J. Wood, "Untethered flight of an insect-sized flapping-wing microscale aerial vehicle," *Nature*, vol. 570, no. 7762, pp. 491–495, Jun. 2019, doi:10.1038/s41586-019-1322-0.
- [3] H. V. Phan, S. Aurecianus, T. K. L. Au, T. Kang, and H. C. Park, "Towards the Long-Endurance Flight of an Insect-Inspired, Tailless, Two-Winged, Flapping-Wing Flying Robot," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5059–5066, Oct. 2020, doi:10.1109/lra.2020.3005127.
- [4] J. Winslow, H. Otsuka, B. Govindarajan, and I. Chopra, "Basic Understanding of Airfoil Characteristics at Low Reynolds Numbers (104–105)," *Journal of Aircraft*, vol. 55, no. 3, pp. 1050–1061, May 2018, doi: 10.2514/1.c034415.
- [5] N. Varsha, M. D. Deshpande, and M. Sivapragasam, "Airfoil Optimisation at Ultra-Low Reynolds Number," *J. Aerosp. Sci. Technol.*, pp. 347–353, 2019.
- [6] J. H. García-Ortiz, A. Domínguez-Vázquez, J. J. Serrano-Aguilera, L. Parras, and C. del Pino, "A complementary numerical and experimental study of the influence of Reynolds number on theoretical models for wingtip vortices," *Computers & Fluids*, vol. 180, pp. 176–189, Feb. 2019, doi: 10.1016/j.compfluid.2018.12.009.
- [7] L. Victoria Rolandi, T. Jardín, J. Fontane, J. Gressier, and L. Joly, "Stability of the Low Reynolds Number Compressible Flow Past a NACA0012 Airfoil," *AIAA Journal*, vol. 60, no. 2, pp. 1052–1066, Feb. 2022, doi: 10.2514/1.j060792.
- [8] T. Kouser, Y. Xiong, D. Yang, and S. Peng, "Direct Numerical Simulations on the three-dimensional wake transition of flows over NACA0012 airfoil at  $Re = 1000$ ," *International Journal of Micro Air Vehicles*, vol. 13, p. 175682932110556, Jan. 2021, doi:10.1177/17568293211055656.

- [9] D. F. Kurtulus, "Unsteady aerodynamics of a pitching NACA 0012 airfoil at low Reynolds number," *International Journal of Micro Air Vehicles*, vol. 11, p. 175682931989060, Jan. 2019, doi:10.1177/1756829319890609.
- [10] K. Suzuki, K. Minami, and T. Inamuro, "Lift and thrust generation by a butterfly-like flapping wing-body model: immersed boundary-lattice Boltzmann simulations," *Journal of Fluid Mechanics*, vol. 767, pp. 659–695, Feb. 2015, doi: 10.1017/jfm.2015.57.
- [11] K. V. Sharma, R. Straka, and F. W. Tavares, "Current status of Lattice Boltzmann Methods applied to aerodynamic, aeroacoustic, and thermal flows," *Progress in Aerospace Sciences*, vol. 115, p. 100616, May 2020, doi: 10.1016/j.paerosci.2020.100616.
- [12] L. Wang et al., "Accurate Computation of Airfoil Flow Based on the Lattice Boltzmann Method," *Applied Sciences*, vol. 9, no. 10, p. 2000, May 2019, doi: 10.3390/app9102000.
- [13] J. A. Reyes Barraza and R. Deiterding, "Towards a generalised lattice Boltzmann method for aerodynamic simulations," *Journal of Computational Science*, vol. 45, p. 101182, Sep. 2020, doi:10.1016/j.jocs.2020.101182.
- [14] N. Pellerin, S. Leclaire, and M. Reggio, "An interpolation-based lattice Boltzmann method for non-conforming orthogonal meshes," *Computers & Mathematics with Applications*, vol. 100, pp. 152–166, Oct. 2021, doi: 10.1016/j.camwa.2021.09.002.
- [15] D. D. Ganji and S. H. H. Kachapi, "Natural, Mixed, and Forced Convection in Nanofluid," *Application of Nonlinear Systems in Nanomechanics and Nanofluids*, pp. 205–269, 2015, doi:10.1016/b978-0-323-35237-6.00006-6.
- [16] D. Solomatine, L. M. See, and R. J. Abraham, "Data-driven modelling: concepts, approaches and experiences," *Pract. hydroinformatics Comput. Intell. Technol. Dev. water Appl.*, pp. 17–30, 2008.
- [17] K. Yonekura and K. Suzuki, "Data-driven design exploration method using conditional variational autoencoder for airfoil design," *Struct. Multidiscip. Optim.*, vol. 64, no. 2, pp. 613–624, 2021.
- [18] X. Hui, J. Bai, H. Wang, and Y. Zhang, "Fast pressure distribution prediction of airfoils using deep learning," *Aerosp. Sci. Technol.*, vol. 105, p. 105949, 2020.
- [19] G. Yao, T. Lei, and J. Zhong, "A review of convolutional-neural-network-based action recognition," *Pattern Recognit. Lett.*, vol. 118, pp. 14–22, 2019.
- [20] L. Alzubaidi et al., "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. big Data*, vol. 8, pp. 1–74, 2021.
- [21] A. Ajit, K. Acharya, and A. Samanta, "A review of convolutional neural networks, p 1--5," in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, IEEE, New York, NY. <https://doi.org/10, 2020>.
- [22] M. Elsaadouny, J. Barowski, and I. Rolfes, "Extracting the features of the shallowly buried objects using LeNet convolutional network," in *2020 14th European conference on antennas and propagation (EuCAP)*, 2020, pp. 1–4.
- [23] Z. Yuan, Y. Wang, Y. Qiu, J. Bai, and G. Chen, "Aerodynamic coefficient prediction of airfoils with convolutional neural network," in *The Proceedings of the 2018 Asia-Pacific International Symposium on Aerospace Technology (APISAT 2018) 9th*, 2019, pp. 34–46.
- [24] 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.
- [25] J. 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, p. 104645, 2020.
- [26] H. Chen, L. He, W. Qian, and S. Wang, "Multiple aerodynamic coefficient prediction of airfoils using a convolutional neural network," *Symmetry (Basel)*, vol. 12, no. 4, p. 544, 2020.
- [27] A. A. Zikri, H. Defiant, W. Hidayat, and A. Purqon, "Geometry Representation Effectiveness in Improving Airfoil Aerodynamic Coefficient Prediction with Convolutional Neural Network," *JOIV Int. J. Informatics Vis.*, vol. 7, no. 3, pp. 644–650, 2023.
- [28] O. R. Bingol and A. Krishnamurthy, "NURBS-Python: An open-source object-oriented NURBS modeling framework in Python," *SoftwareX*, vol. 9, pp. 85–94, 2019.
- [29] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*, 2017, pp. 1–6.
- [30] S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik, "Prediction of aerodynamic flow fields using convolutional neural networks," *Comput. Mech.*, vol. 64, pp. 525–545, 2019.
- [31] P. L. Bhatnagar, E. P. Gross, and M. Krook, "A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems," *Phys. Rev.*, vol. 94, no. 3, p. 511, 1954.
- [32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [34] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315–323.
- [35] 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: Proceedings of MARC 2018*, 2019, pp. 873–880.