

## INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION



journal homepage: www.joiv.org/index.php/joiv

# Geometry Representation Effectiveness in Improving Airfoil Aerodynamic Coefficient Prediction with Convolutional Neural Network

Arizal Akbar Zikri<sup>a,b</sup>, Hanni Defianti<sup>c</sup>, Wahyu Hidayat<sup>a</sup>, Acep Purqon<sup>a,\*</sup>

<sup>a</sup> Department of Computational Science, Bandung Institute of Technology, Bandung, Indonesia <sup>b</sup> National Laboratory for Aerodynamics, Aeroelastics, and Aeroacoustics Technology, National Research and Innovation Agency, Tangerang Selatan, Indonesia <sup>c</sup> Aeronautics Research Center, National Research and Innovation Agency, Bogor, Indonesia

Corresponding author: \*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. Obtaining the aerodynamic coefficients is a must to optimize the airfoil design. Engineers use various methods to get the airfoil aerodynamic coefficients. A prediction method is an approximation approach that effectively reduces time and cost. This article uses convolutional neural networks (CNN) to get approximation values of those coefficients. In CNN, we collect 8920 aerodynamic coefficients for 223 NACA 4 as labels in datasets by using XFOIL at  $M_a = 0$  and  $R_e = 500000$  with varying angles of attacks starting  $-20^\circ$  to  $20^\circ$  with increment of  $1^\circ$ . The simulation results are compared to the experiment using E387 airfoil for validation. Then, airfoil geometries as part of input datasets were transformed into Grayscale and RGB images using the signed distance function (SDF) and mesh algorithm. Each airfoil representation was trained using an 80% dataset and tested using a 20% dataset with Adam as an optimizer to generate each prediction model using modified LeNet-5. We use three different layer depths in modified LeNet-5 to obtain the optimal layer number. There is no remarkable improvement when varying the depth layers, so four layers are used instead. Simulation results show that using an SDF with Fast Marching Method on CNN predicts the most effective for the airfoil's lift, drag, and pitch moment coefficient with varying angles of attack simultaneously. One can extend the method by using SDF to recognize different flow conditions.

Keywords- Deep neural network; CNN; airfoil; aerodynamic coefficient; prediction.

Manuscript received 14 Jan. 2023; revised 25 Apr. 2023; accepted 9 May 2023. Date of publication 10 Sep. 2023. International Journal on Informatics Visualization is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



## I. 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 airfoil sections [1]. Many airfoils have interesting aerodynamic behaviors that could impact flight safety and performance. Therefore, Obtaining the aerodynamic coefficients is a must [2]. These aerodynamic coefficients have received much attention in experiments through wind tunnel tests 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 [3]. Due to recent technology in numerical computation, engineers use various methods to compute aerodynamic coefficients as the first step in airfoil design [4]; then, wind tunnel testing can validate the result of the aerodynamic coefficient.

He et al. [4] use a numerical approach for airfoil design. Another strategy is using Particle Swarm Optimization (PSO) to produce an optimal airfoil design [5]. A spotted hyena optimizer, a metaheuristic approach, is used to optimize airfoil design by minimizing the drag force [6]. 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 to solve that situation [7].

Nowadays, data size is growing unprecedentedly [8]-[10], so airfoil design and analysis based on data-driven is

preferable [11]-[14]; moreover, the rise of computer technology and Artificial Intelligence (AI) has guided the reliable evolution of the data-driven model. Yonekura et al. [15] dan Thuerey et al. [16] utilize the Autoencoder method to explore airfoil design. Sekar et al. [17] 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 realized as deep learning and utilizes convolution segments to gather specific patterns from pictures and use them for training and validation. Zhang et al. [18] approximate the lift force coefficient with CNN using an airfoil geometry-image as input to the trained model. Conversely, Chen et al. [19] use CNN to get the lift, drag, and moment predictions. Hui et al. [20] utilize CNN to predict the pressure distribution around the airfoil surface. All previous studies need airfoil geometry representations as input in CNN.

There are several methods to generate airfoil geometry representations. Many airfoil geometry representation methods use various types of aspects. The grayscale image's outline and solid color represent the airfoil geometry [18], [19]. The grayscale image only has one channel, which has less pixel information and affects image recognition.

We also use Unstructured Mesh (UM) as an airfoil geometry representation based on a meshing algorithm [21] to enrich the pixel information on the grayscale image. However, UM image algorithm randomly creates patterns, as shown in Fig. 2b, making it challenging to recognize image features. To overcome the shortcomings of Grayscale and UM image, we propose the enhanced Signed Distance Function (SDF) method [20], [22] through the Fast Marching Method (FMM). SDF+FMM uses three channels to enrich the pixel information. In addition, SDF+FMM produces more consistent patterns than UM with faster performance and improves aerodynamic coefficient prediction.

## II. MATERIAL AND METHOD

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



Fig. 1 Predictive model development flowchart

#### A. Airfoil Data Processing

The first step in processing airfoil data in building a predictive model is obtaining the airfoil's shape or geometry. The airfoil shape consists of coordinates along the top and bottom surfaces. The coordinates are acquired from UIUC Airfoil Data Site and converted into a digital image, so it is easy to implement the architecture with CNN [23]. Airfoil images can be created in the form of a single channel [18], [19], [24] [20], [25] using a Signed Distance Field [26].



(a) Grayscale image

(b) MU image



Fig. 2 Various airfoil geometry representations

Airfoil geometry representation with Grayscale (Fig. 2a) is our base representation and is easy to generate. It contains only black and white pixels. We also use UM (Fig. 2b) to enrich the pixel information from the Grayscale. Then for comparison, we propose SDF+FMM (Fig. 2c) for airfoil geometry representation based on Algorithm 1. The SDF+FMM can give more complex features than Grayscale but has a more precise pattern than UM. The images and the airfoil aerodynamic coefficients are stored in the database.

We use XFOIL [27] to obtain the airfoil aerodynamic coefficients. XFOIL can predict airfoil aerodynamics fastly for a low Reynold number [28]-[30] 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  $(C_l, C_d, and C_m)$  with varying angles of attack, starting  $-20^{\circ}$  until  $20^{\circ}$ . Finally, we keep the airfoil surface roughness configuration  $N_{crit} = 5$ .

Algorithm 1: The proposed algorithm for implementing SDF+FMM					
Data: Airfoil geometries					
Result: Airfoil images					
$foils \leftarrow geometries;$					
for $i \leftarrow 0$ to $len(foils)$ by 1 do					
if $foils[i].type == "lednicer"$ then					
$foils[i] \leftarrow MoveFirstHalfRow(foils[i].data) // Transform to Selig$					
end					
for $j \leftarrow 0$ to $len(foils[i].data)$ by 1 do					
$points \leftarrow foils[i].data[j];$					
$points \leftarrow points * scale ;$	// Scale the points				
$img \leftarrow DrawAirfoilImg(points);$	// Draw the grayscale images				
$img \leftarrow GenerateSDFFMM(img);$	// Generate the SDF+FMM images				
SaveImg(img);					
end					
end					

To validate the computation results using XFOIL, we compare XFOIL and the experiment [31] using the same E387 airfoil. It turns out that the results between them are acceptable based on Fig. 3.



Fig. 3 XFOIL and experiment results

## B. CNN Architecture

CNN has been widely used in image recognition and was first proposed for gradient-based learning in document recognition [32]. CNN is an ideal architecture for image data implementation [33]. 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. 4, we use LeNet-5 [34] as CNN architecture with some adjustments. The airfoil image has a size of  $128 \times 128$  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  $15 \times 15$  as in Fig. 4b or Fig. 4c, with three channels as input and 1 step size in each direction. We use a batch norm [35], [36] to make it faster and more stable during training. The activation function is a rectified linear unit (ReLU) because it gives better results [37]–[39].





After the ReLU operation in Fig. 4c, we set 20 channels with a resolution of  $58 \times 58$  after the MaxPool procedure based on Eq. (1 in the article [40] for the Conv layer 2. We can get output resolution (*o*) with *IN* N as input, *PD* as padding, *KN* as kernel, and *ST* as stride (step) size

$$o = \left[\frac{IN + 2PD - KN}{ST}\right] + 1 \tag{1}$$

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 the output resolutions are shown in Tabel. The output layer of this architecture is linear regression. It uses MSE (Mean Square Error) as a loss

function to measure the performance of the trained model to know how good it is at predicting the coefficients.

TABLE I
THE OUTPUT RESOLUTIONS FROM THE LAST CONV LAYER INTO A FULLY
CONNECTED LAYER

Layer	Aerofoil2BN2FC	Aerofoil3BN2FC	Aerofoil4BN2FC
CONV1	$32 \times 28 \times 28$	16 × 57 × 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	-	-	$60 \times 4 \times 4$

## C. Model Training

The model training includes two steps: forward and backward calculation. The first step is to extract the airfoil image patterns using the convolution and pooling procedures, feed the components into the fully connected segment, and gets aerodynamic coefficient prediction results through the output layer. Prediction errors are the differences between approximation and actual values. In the last step, the algorithms return these errors and update the network parameters. The numbers of convolutional, pooling, and fully connected layers are flexible depending on the input images.

The forward and backward calculation steps are repeated if the stop condition is unmet. The maximum number of epochs determines the end condition of CNN training. The training procedure of the CNN architecture is shown in Fig. 5.



Fig. 5 The proposed procedure for model training

In this paper, the training algorithm uses adaptive moment estimation (Adam) [41], [42]. It minimizes the loss criterion using the minibatch. An epoch is the complete progress of the instruction rules over the whole training set.

## III. RESULTS AND DISCUSSION

We collect 223 NACA 4 airfoils and have 8920 in total datasets. The datasets include airfoil images as input and aerodynamic 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, and then the pictures and airfoil coefficients are standardized using a mean ( $\mu$ ) of 0.5 and a standard deviation ( $\sigma$ ) of 0.5.

Fig. 6 shows the training history for various airfoil geometry representations using the same Aerofoil4BN2FC architecture as shown in Fig. 4c. 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.



Fig. 6 Training history for various airfoil geometry representations

Because the SDF+FMM can perform very well during training, we combine it with multiple architectures, as shown in Fig. 4. There is no significant improvement when varying the convolution layers while validating the datasets. However, all architectures show decreasing MSE during training, as shown in Fig. 7, so we choose the four layers instead (Aerofoil4BN2FC).



Fig. 7 Training history for layer depth variation on SDF datasets

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 prediction results. From Fig. 8a, the image with SDF+FMM has the best prediction result. The grayscale image has a better prediction result than the unstructured mesh. The unstructured mesh cannot perform well at less than zero angles of attack. For the drag coefficient case in Fig. 8b, the unstructured mesh performs worst for all conditions. The grayscale image performs well, but the SDF+FMM image still has the best prediction results. It also happens on the moment coefficient, as shown in Fig. 8c, and the SDF+FMM 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.

The SDF+FMM image can perform very well for aerodynamic coefficient. This performance can happen because the trained model using SDF+FMM as airfoil geometry representation has the smallest MSE during validation. In this article, 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 our proposed SDF+FMM has the best metric results compared to the original SDF.



THE METRIC RESULTS FROM VARIOUS GEOMETRY REPRESENTATIONS

	RMSE		$r^2$			
Geometry	$C_l$	C <sub>d</sub>	$C_m$	$C_l$	$C_d$	$C_m$
Grayscale	0.0336	0.0067	0.0126	0.9994	0.9822	0.9629
ŬM	0.2337	0.0214	0.0192	0.9848	0.8828	0.8227
SDF+FMM	0.0245	0.0036	0.0032	0.9994	0.9952	0.9946
Composite [19]	0.0273	0.0035	0.0027	-	-	-
SDF [25]	0.0296	0.0042	0.0042	-	-	-

In terms of the  $r^2$  metric, as illustrated in Fig. 9, the grayscale image and SDF+FMM 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 Fig. 8a, but it has a high value of RMSE, and 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+FMM can fit close enough to actual  $C_d$  as shown Fig. 8b. The Grayscale has a lower RMSE than the UM. The Grayscale and UM have higher RMSE than the SDF+FMM for predicting  $C_m$ .



Fig. 9 The  $r^2$  metric for various prediction result

#### **IV. CONCLUSIONS**

Many options, including Grayscale, UM, and SDF, can be used as airfoil geometry representations. This choice determines the level of effectiveness of the CNN model during training. In the training process, the CNN model uses a modified LeNet-5 architecture. This architecture has three depth layers, Aerofoil2BN2FC, Aerofoil3BN2FC, and Aerofoil4BN2FC. Based on this study, we use 8K datasets, and the Aerofoil4BN2FC architecture type was chosen to determine which image type is better Grayscale, UM, or SDF+FMM. Fig. 6 shows that SDF+FMM images give the best results during the training process at 1000 epochs because it produces more consistent patterns and more complex pixel information. We use Batch Normalization to speed up the training process. The training process and the metric results also show that the image we propose in SDF+FMM has the smallest RMSE value of 0.0245 for  $C_l$ , 0.0036 for  $C_d$ , and 0.0032 for  $C_m$ , than the original SDF. The same applies to the  $r^2$  value of 99.9% for  $C_l$ , 99.5% for  $C_d$ , and 99.46% for  $C_m$ , in predicting the airfoil aerodynamic coefficient according to Table II. This study also found variations in the depth of the CNN layer on SDF+FMM images have a similar performance during training, so the architecture chosen is Aerofoil4BN2FC.

The SDF+FMM image effectively improves training performance and predicts aerodynamic coefficients very well. SDF+FMM images can be applied to airfoils (2D wings) on different flow conditions for further research. But still, this image needs improvement for predictions of the aerodynamic coefficient on 3D wings. However, this limitation opens the opportunity for further research on predicting aerodynamic coefficients using CNN for airfoils and 3D wings with different flow conditions.

### ACKNOWLEDGMENT

The computation in this work has been done using the facilities of MAHAMERU BRIN HPC, National Research and Innovation Agency of Indonesia (BRIN).

#### REFERENCES

- H. Zhao, Z. Gao, Y. Gao, and C. Wang, "Effective robust design of [1] high lift NLF airfoil under multi-parameter uncertainty," Aerosp Sci vol. 68, 530-542, Technol, Sep. 2017. doi: pp. 10.1016/j.ast.2017.06.009.
- P. D. Bravo-Mosquera, H. D. Cerón-Muñoz, G. Díaz-Vázquez, and F. [2] Martini Catalano, "Conceptual design and CFD analysis of a new prototype of agricultural aircraft," Aerosp Sci Technol, vol. 80, pp. 156-176, Sep. 2018, doi: 10.1016/j.ast.2018.07.014.
- J. Tu, G. H. Yeoh, and C. Liu, Computational fluid dynamics: a [3] practical approach. Butterworth-Heinemann, 2018.
- X. He, J. Li, C. A. Mader, A. Yildirim, and J. R. R. A. Martins, [4] "Robust aerodynamic shape optimization-from a circle to an airfoil," Aerosp Sci Technol, vol. 87, pp. 48-61, 2019.
- S. Mirjalili, J. S. Dong, A. Lewis, and A. S. Sadiq, "Particle swarm [5] optimization: theory, literature review, and application in airfoil design," Nature-inspired optimizers, pp. 167-184, 2020.
- G. Dhiman and A. Kaur, "Optimizing the design of airfoil and optical [6] buffer problems using spotted hyena optimizer," Designs (Basel), vol. 2, no. 3, p. 28, 2018.
- B. Crawford et al., "Q-learnheuristics: towards data-driven balanced [7] metaheuristics," *Mathematics*, vol. 9, no. 16, p. 1839, 2021. Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning
- [8] for big data," Information Fusion, vol. 42, pp. 146-157, 2018.
- [9] A. Oussous, F.-Z. Benjelloun, A. Ait Lahcen, and S. Belfkih, "Big Data technologies: A survey," Journal of King Saud University Computer and Information Sciences, vol. 30, no. 4, pp. 431-448, Oct. 2018, doi: 10.1016/j.jksuci.2017.06.001.
- [10] Y. Duan, J. S. Edwards, and Y. K. Dwivedi, "Artificial intelligence for decision making in the era of Big Data – evolution, challenges and research agenda," *Int J Inf Manage*, vol. 48, pp. 63–71, Oct. 2019, doi: 10.1016/j.ijinfomgt.2019.01.021.
- [11] E. Andrés-Pérez, "Data Mining and Machine Learning Techniques for Aerodynamic Databases: Introduction, Methodology and Potential Benefits," Energies (Basel), vol. 13, no. 21, p. 5807, 2020.
- [12] P. Gupta, P. Tyagi, and R. K. Singh, "Analysis of Generative Adversarial Networks for Data-Driven Inverse Airfoil Design," 2022, pp. 251-261. doi: 10.1007/978-981-16-7618-5\_22.
- [13] J.-Y. Li, Z.-H. Zhan, H. Wang, and J. Zhang, "Data-Driven Evolutionary Algorithm With Perturbation-Based Ensemble Surrogates," IEEE Trans Cybern, vol. 51, no. 8, pp. 3925-3937, Aug. 2021, doi: 10.1109/TCYB.2020.3008280.
- [14] J. Li, S. He, and J. R. R. A. Martins, "Data-driven constraint approach to ensure low-speed performance in transonic aerodynamic shape optimization," Aerosp Sci Technol, vol. 92, pp. 536-550, Sep. 2019, doi: 10.1016/j.ast.2019.06.008.
- [15] K. Yonekura and K. Suzuki, "Data-driven design exploration method using conditional variational autoencoder for airfoil design," Structural and Multidisciplinary Optimization, vol. 64, no. 2, pp. 613-624, 2021.
- [16] N. Thuerey, K. Weißenow, L. Prantl, and X. Hu, "Deep learning methods for Reynolds-averaged Navier-Stokes simulations of airfoil flows," AIAA Journal, vol. 58, no. 1, pp. 25-36, 2020.
- V. Sekar, M. Zhang, C. Shu, and B. C. Khoo, "Inverse design of airfoil [17] using a deep convolutional neural network," Aiaa Journal, vol. 57, no. 3, pp. 993–1003, 2019.
- Y. Zhang, W. J. Sung, and D. N. Mavris, "Application of [18] convolutional neural network to predict airfoil lift coefficient," in 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2018, p. 1903.
- H. Chen, L. He, W. Qian, and S. Wang, "Multiple aerodynamic [19] coefficient prediction of airfoils using a convolutional neural network," Symmetry (Basel), vol. 12, no. 4, p. 544, 2020.
- X. Hui, J. Bai, H. Wang, and Y. Zhang, "Fast pressure distribution prediction of airfoils using deep learning," *Aerosp Sci Technol*, vol. [20] 105, p. 105949, 2020.
- [21] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities," Int J Numer Methods Eng, vol. 79, no. 11, pp. 1309-1331, 2009.

- [22] Z. Yang, J. Wu, P. M. Thompson, and Y. Wang, "Deep Learning on SDF for Classifying Brain Biomarkers," in 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), IEEE, Nov. 2021, pp. 1051-1054. doi: 10.1109/EMBC46164.2021.9630850.
- [23] 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.
- J. Viquerat and E. Hachem, "A supervised neural network for drag [24] prediction of arbitrary 2D shapes in laminar flows at low Reynolds number," Comput Fluids, vol. 210, p. 104645, 2020.
- [25] 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, 2018, pp. 34-46.
- S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik, [26] "Prediction of aerodynamic flow fields using convolutional neural networks," Comput Mech, vol. 64, no. 2, pp. 525-545, 2019.
- M. Drela, "XFOIL: An analysis and design system for low Reynolds [27] number airfoils," in Low Reynolds number aerodynamics, Springer, 1989, pp. 1-12.
- J. Morgado, R. Vizinho, M. A. R. Silvestre, and J. C. Páscoa, "XFOIL [28] vs CFD performance predictions for high lift low Reynolds number airfoils," Aerosp Sci Technol, vol. 52, pp. 207-214, 2016.
- [29] C. Echavarria, J. D. Hoyos, J. H. Jimenez, G. Suarez, and A. Saldarriaga, "Optimal airfoil design through particle swarm optimization fed by CFD and XFOIL," Journal of the Brazilian Society of Mechanical Sciences and Engineering, vol. 44, no. 11, p. 561, Nov. 2022, doi: 10.1007/s40430-022-03866-4.
- [30] A. Cruz et al., "Aerodynamic Performance of Symmetric and Cambered Airfoils at Low Reynolds Numbers," in Volume 1: Fluid Applications and Systems (FASTC); Fluid Measurement and Instrumentation (FMITC); Fluid Mechanics (FMTC), American Society of Mechanical 2022. Engineers, Aug. doi: 10.1115/FEDSM2022-86960.
- M. S. Selig and J. J. Guglielmo, "High-lift low Reynolds number [31] airfoil design," J Aircr, vol. 34, no. 1, pp. 72-79, 1997.
- Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based [32] learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.
- [33] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016
- B. Yu, L. Xie, and F. Wang, "An Improved Deep Convolutional [34] Neural Network to Predict Airfoil Lift Coefficient," 2020, pp. 275-286. doi: 10.1007/978-981-15-1773-0 21.
- C. Garbin, X. Zhu, and O. Marques, "Dropout vs. batch normalization: [35] an empirical study of their impact to deep learning," Multimed Tools Appl, vol. 79, no. 19-20, pp. 12777-12815, May 2020, doi: 10.1007/s11042-019-08453-9
- V. Thakkar, S. Tewary, and C. Chakraborty, "Batch Normalization in [36] Convolutional Neural Networks - A comparative study with CIFAR-10 data," in 2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT), IEEE, Jan. 2018, pp. 1-5. doi: 10.1109/EAIT.2018.8470438.
- X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural [37] networks," in Proceedings of the fourteenth international conference on artificial intelligence and statistics, 2011, pp. 315-323.
- [38] 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.
- X. Jiang, Y. Pang, X. Li, J. Pan, and Y. Xie, "Deep neural networks [39] with Elastic Rectified Linear Units for object recognition," *Neurocomputing*, vol. 275, pp. 1132–1139, Jan. 2018, doi: 10.1016/j.neucom.2017.09.056.
- V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep [40] learning," arXiv preprint arXiv:1603.07285, 2016.
- K. Bae, H. Ryu, and H. Shin, "Does Adam optimizer keep close to the [41] optimal point?," arXiv preprint arXiv:1911.00289, 2019.
- S. Bock and M. Weis, "A Proof of Local Convergence for the Adam [42] Optimizer," in 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, Jul. 2019, 1 - 8. pp. doi: 10.1109/IJCNN.2019.8852239.