Transport and Communications Science Journal

CONVOLUTIONAL NEURAL NETWORK FOR DETERMINING THE FLOW FIELD AROUND AN AIRFOIL AND BLUNT-BASED MODELS

Tran The Hung[*]

Faculty of Aerospace Engineering, Le Quy Don Technical University, No 236 Hoang Quoc Viet Street, Bac Tu Liem, Hanoi, Vietnam

ARTICLE INFO

TYPE: Research Article
Received: 05/06/2024
Revised: 24/10/2024
Accepted: 10/01/2025
Published online: 15/01/2025
https://doi.org/10.47869/tcsj.76.1.3

[*] *Corresponding author*
Email: tranthehung_k24@lqdtu.edu.vn; Tel: +84355544745

Abstract. The convolutional neural network is widely applied in the classification of images and medicine. Some current networks are used in aerospace engineering and show a high potential in determining aerodynamic forces and flow fields. This article constructs a convolutional neural network for predicting pressure and velocity fields around a two-dimensional aircraft wing model (airfoil model). Training data is computed using the Reynolds-averaged method, and then extracted, focusing on the flow around the wing. Input data includes geometric parameters, and airfoil inlet velocity, and output data includes pressure field and flow velocity around the airfoil. The convolutional neural network is based on improving the U-Net network model, commonly used in medical applications. The results show that the convolutional neural network accurately predicts flow around the airfoil, with an average error below 3%. Therefore, this network can be used and further developed to predict flow around the wing. The network is then applied to predict the pressure and pressure fields around a blunt-based model with different aspect ratios. The main feature of the flow can be extracted from the network. Results related to pressure distribution, velocity, and method error are presented and discussed in the study. This study also suggests improving the network and applying it to pressure and velocity fields in aerospace engineering.

Keywords: Convolutional neural network, U-Net, airfoil, blunt-based model, flow fields, pressure, velocity.

1. INTRODUCTION

Over the past century, the aircraft wing has remained a pivotal component for generating lift, which is crucial for flight in the atmosphere. With advancements in aviation and computational methodologies, wing database systems have been continually enhanced. Prominent examples include the NACA wing system and Xfoil software, which swiftly provide data on lift, drag, and pressure distribution on the wing surface [1]. More precise techniques, such as utilizing software to tackle finite volume problems, enable comprehensive analysis of pressure, velocity, and friction around the wing surface, rooted in classical mathematical equations and computational space discretization [2,3].

In contemporary times, artificial neural networks have gained widespread application across scientific and engineering domains. Leveraging substantial training datasets, neural networks yield predictions with minimal errors compared to conventional methods. In aerodynamics, artificial neural networks are employed to forecast the lift and drag values of models. Global networks like convolutional neural networks facilitate the precise distribution of pressure and velocity fields around models with minimal errors. This methodology involves generating training and testing data from traditional computational approaches, and then restructuring them into four-dimensional arrays. Subsequently, the data undergoes convolutional neural network processing to extract features, which are then reconstructed into pressure and velocity fields. Throughout the training phase, network parameters are fine-tuned to yield pressure and velocity field results resembling the original data. Various networks have been developed for this task, such as FlowNet for optical flow and U-Net for medical applications [4]. However, it's worth noting that artificial neural networks, beyond their mathematical foundations, possess architectural nuances, influencing the outcome based on the chosen convolutional network design.

Another difficult task is from prediction of the flow field around the blunt base model. This model features a large separation flow at the base, which results in a low-velocity region and high aerodynamic drag [5]. Building a network for prediction flow around the blunt body is also an important task, which was not been conducted before. It is also interesting to know how much is the accuracy of the current network in predicting the pressure and velocity fields around the blunt-base model.

In this research, we suggest alterations to the conventional U-Net architecture to facilitate the extraction of pressure and velocity fields surrounding a two-dimensional aircraft wing model (airfoil). Our dataset comprises 400 airfoil instances with varied shapes and flow conditions for both training and testing. Our findings from training the U-Net model demonstrate its capability to predict velocity and pressure field characteristics with high accuracy, exhibiting a typical error rate of under 3%. Thus, these modified networks hold promise for computational fluid dynamics applications concerning physical model analysis.

2. CONVOLUTIONAL NEURAL NETWORK DIAGRAM AND TRAINING DATA

2.1. Convolutional neural network diagram

U-Net Convolutional Neural Network is a network architecture used in the field of image processing, particularly for segmentation tasks [4] [6] [7]. It was also used by Wu et al. [8] for detection of the infrared object. This architecture is designed to retain high-level information (learned from convolutional layers) while also maintaining specific positional information (learned from pooling layers). U-Net is typically divided into two main parts: the encoder and the decoder. The encoder uses convolutional layers to extract information from the input image and applies pooling layers to reduce the feature size while retaining important information. Conversely, the decoder uses transposed convolutional layers to reconstruct the image with high resolution and combines information from the corresponding encoding layers through skip connections to recreate specific objects. U-Net has demonstrated good performance in various applications, including cell segmentation in medical images, object recognition in images, and many other tasks. The unique structure of U-Net allows it to retain both high-level and positional information, making it a popular choice for tasks that require both detailed and positional information about objects. The output results depend on the number of layers in the U-Net. For the airfoil models, this study uses a U-Net with three input-output layers. Additionally, in comparison to the previous study by Du et al [4], the network is redesigned by us for a suitable application of the airfoil model. By reducing the number of layers, the parameters of the network are reduced and the training process becomes faster. The network input is modified to be a three-dimensional matrix of size 128×128×3. The size of the image is also modified for the current study. The first two dimensions represent the image size, and the third dimension sequentially represents the model's geometry, the input velocity in the *x* direction, and the input velocity in the *y* direction. Each convolutional layer is followed by a ReLU layer, and the final convolutional layer is followed by a Max Pooling layer. Initially, the U-Net network was used for image segmentation. Therefore, the output parameter is changed to a three-dimensional matrix of size 128×128×3, where the third dimension sequentially represents the pressure field, the output velocity in the *x* direction, and the output velocity in the y direction. The network includes 7.6 million parameters. The structure of the U-Net and its parameters are presented in Figure 1. The network structure and training process are built using MATLAB software. Previously, Du et al [4] used Python and many processes for the training. In this study, we create a program, so all parameters and their effect on the results can be controlled.

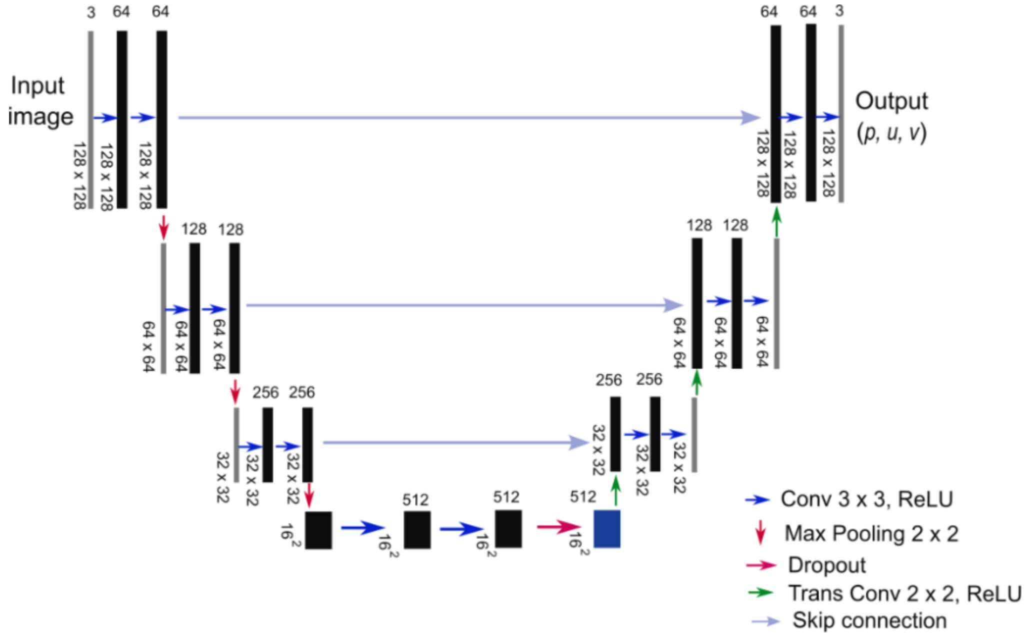

Figure 1. Diagram of the 3-layer U-Net network.

2.2. Training Data

The training data used in this study is taken from the dataset published by Thuerey and colleagues [6]. Specifically, the Reynolds-Averaged Navier-Stokes (RANS) method with the Spalart-Allmaras turbulence model is used. The model was developed by Spalart and Allmaras [9], which uses one additional equation for predicting turbulent eddy viscosity. The equation for the model is shown in Eq. (1). The biggest advantage of the Spalart-Allmaras model is that the simulation is fast and the requirement of y^+ is not strong. Consequently, it is widely used in fluid mechanics in the initial aerodynamic designing process [10].

$$\frac{\partial \hat{\nu}}{\partial t}+u_j\frac{\partial \hat{\nu}}{\partial x_j}=c_{b1}\left(1-f_{t2}\right)\hat{S}\hat{\nu}-\left[c_{\omega1}f_{\omega}-\frac{c_{b1}}{\kappa^2}f_{t2}\right]\left(\frac{\hat{\nu}}{d}\right)^2+\frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left(\left(\nu+\hat{\nu}\right)\frac{\partial \hat{\nu}}{\partial x_j}\right)+c_{b2}\frac{\partial \hat{\nu}}{\partial x_i}\frac{\partial \hat{\nu}}{\partial x_i}\right] \quad (1)$$

Where the turbulent eddy viscosity is calculated by:

$\mu_t=\rho\hat{\nu}f_{v1}$ with $f_{v1}=\dfrac{\chi^3}{\chi^3+c_{v1}^3}$ and $\chi=\dfrac{\hat{\nu}}{\nu}$. Here, ρ is air density, v is kinematic viscosity and μ is the dynamic viscosity. Additional definitions are given by the following:

$$\hat{S}=\Omega+\frac{\hat{\nu}}{\kappa^2 d^2}f_{v2} \quad \text{with} \quad \Omega=\sqrt{2W_{ij}W_{ij}} \quad \text{and} \quad f_{v2}=1-\frac{\chi}{1+\chi f_{v1}} ,$$

$$f_{\omega}=g\left[\frac{1+c_{\omega3}^6}{g^6+c_{\omega3}^6}\right]^{1/6} \quad g=r+c_{\omega2}\left(r^6-r\right) , \quad r=\min\left[\frac{\hat{\nu}}{\hat{S}\kappa^2 d^2},10\right] , \quad f_{t2}=c_{t3}\exp(-c_{t4}\chi^2) ,$$

$$W_{ij}=\frac{1}{2}\left[\frac{\partial x_i}{\partial x_j}-\frac{\partial x_j}{\partial x_i}\right]$$

The other constant parameters are selected as:

c_{b1} = 0.1355, σ = 2/3, c_{b2} = 0.622, κ = 0.41, $c_{\omega2}$ = 0.3, $c_{\omega3}$ = 2, c_{v1} = 7.1, c_{t3} = 1.2, c_{t4} = 0.5 and $c_{\omega1}=\dfrac{c_{b1}}{\kappa^2}+\dfrac{1+c_{b2}}{\sigma}$.

Calculations are performed in the OpenFoam environment. The geometric features and the flow around the model are cropped to a size of 128 × 128 pixels to facilitate the training process. Note that since a MATLAB program is used, the size of each node is the same for points close and far from the models. Consequently, the boundary layer cannot be captured for the training data. A total of 400 data sets are used for training. The angle of attack is changed from -22.5° to 22.5°. The Reynolds number is in the range of 0.5-5.0 million. Consequently, separation flow can occur on the surface, and training data contains different flow types. The velocity was then normalized by freestream velocity while the pressure was normalized by static pressure before training. An example of the training data field is shown in Figure 2. Here, the x and y axes present the pixel number and the flow is from the left to right. It includes the geometry, the free stream in x and y directions, the results of pressure fields, and velocity fields. All data of the input and output has the same size of 128 × 128 pixels. Note that this data was generated by Thuerey and colleagues [6] with detailed validation. Here, we can see that the numerical method can simulate well the flow phenomenon around the airfoil model such as the separation flow on the trailing edge, a high pressure and low velocity around the leading edge. However, since the data in MATLAB is organized as a matrix, the boundary layer may not be captured well in the simulation. Consequently, it can be confirmed that the training data is reliability.

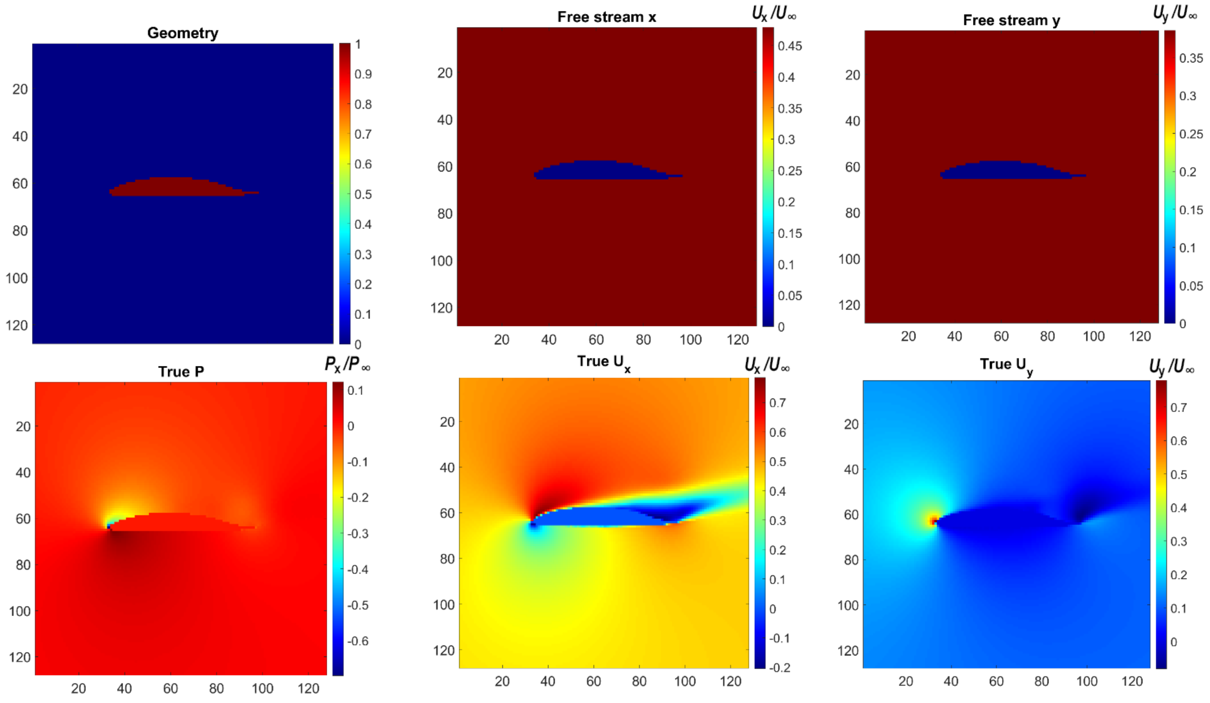

Figure 2. Training data for the training process.

2.3. Training Model

The three-layer U-Net network described in section 2.1 is used for the training process. The airfoil data is divided into 80% for training and 20% for testing. The training data is divided into mini-Batch with a size of 10. The loss function is calculated as the average error of the pressure field and velocities during training with the standard data. The loss function of is determined by Eq. (2):

$$Loss(P_x) = \left\| P_{x,True} - P_{x,Pridict} \right\|$$

$$Loss(V_x) = \left\| V_{x,True} - V_{x,Pridict} \right\| \tag{2}$$

$$Loss(U_x) = \left\| U_{x,True} - U_{x,Pridict} \right\|$$

$$Loss = mean\left(Loss(P) + Loss(U_x) + Loss(U_y) \right)$$

Note that the loss function by Equation (2) was often used for training process of aerodynamic qualities, previously. A total of 100 epochs are performed for the training process. The adaptive moment estimation (Adam) algorithm is used. The learning rate is fixed at 0.001. It should be noted that the learning rate can affect the convergence of the problem. However, calculations in this study show that reducing or changing the learning rate around the chosen parameter does not significantly change the loss function. Additionally, four, and five-layer U-Net were also attempted in predicting the pressure and velocity fields. However, the training time increases while the results are not improved. Consequently, a three-layer network was applied in the current study. The training process is performed using MATLAB software on the graphical card (GPU). For the training, we used our personal computer Lenovo P50 core i7 with four cores, RAM 32Gb and GPU 2Gb. The time for training is around one hour for one case.

3. RESULTS AND DISCUSSION

3.1. Training Error

Figure 3 shows the changes in the loss function over the number of epochs. An enlarge result of loss function is also presented in the right figure. It can be seen that the loss function decreases rapidly at the beginning and gradually decreases up to 3000 iterations. However, when further increasing the number of iterations, the results change little and converge to a value of 0.05. The loss error is small, and the training results for the pressure and velocity fields can be obtained with small errors.

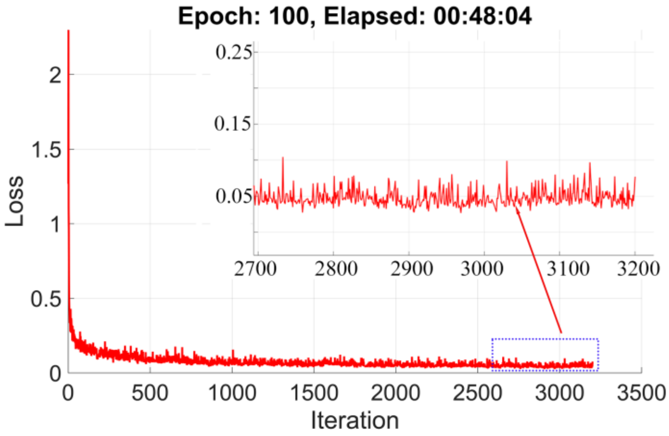

Figure 3. Changes in the loss function over iterations.

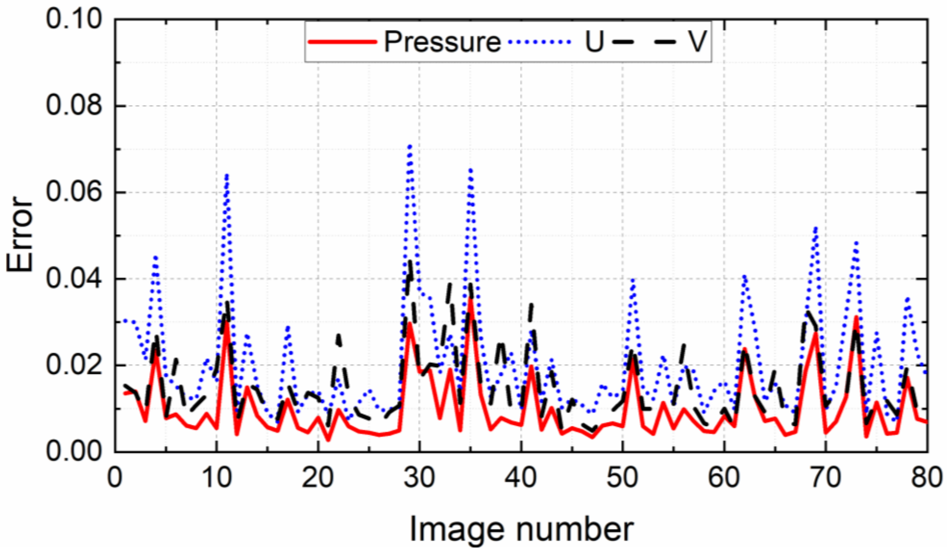

Figure 4. Changes in error over test data.

Figure 4 shows the average error results of the test data. Here, the rest eighty data sets are used for testing. Note that this is the rest of the data, so we observe that the error is not decreased with the number of images. It can be seen that the average error of the calculations is less than 3% for both the pressure field and velocities. However, in some cases, the error increases to 6% or 8%. This can be explained by the fact that when changing the angle of attack, the flow field around the model becomes complex, and thus the error tends to increase. However, the average error is small, indicating that the method is effective in predicting the flow field around the model.

3.2. Training Results

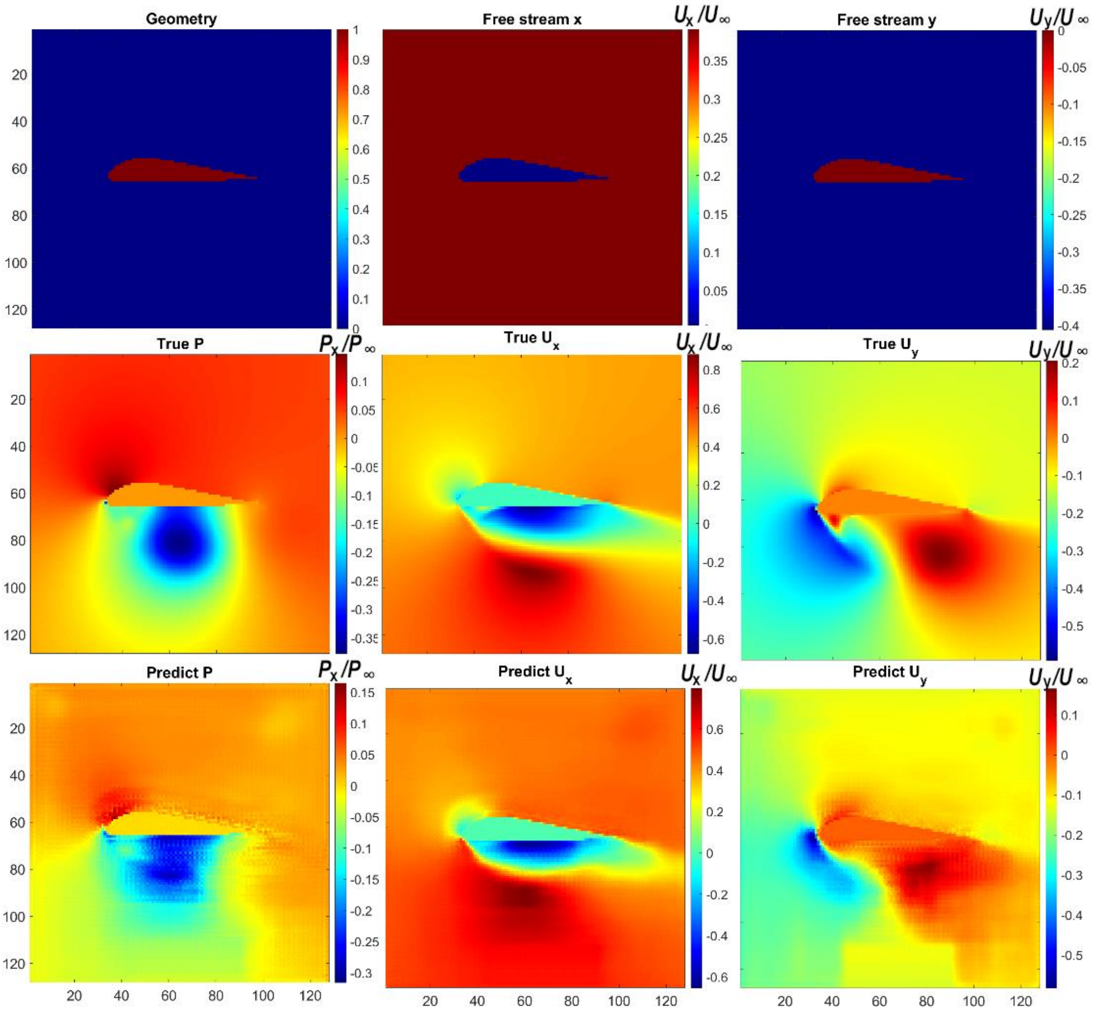

Figure 5. Changes in error over test data.

Figure 5 shows the training results of an airfoil model after 100 epochs. It can be seen that the training results predict the pressure and velocity fields around the model quite accurately. In the details, the low-pressure and low-velocity region below the airfoil can be described quite accurately from the training. Similarly, the high-velocity region above the model can be described relatively accurately through training. Far from the airfoil, the flow velocity turns to the freestream value and can be obtained well from the training process. Consequently, the main characteristics of the pressure and velocity can be captured highly accurately from the training results. A limitation is that in comparison to numerical simulation calculations, the training results show less smoothness, especially in the pressure field on the upper surface and velocity field *v*. This can be explained by the fact that the training model considers each pixel individually and lacks mathematical connections between neighboring pixels.

3.3. Flow around thin airfoils

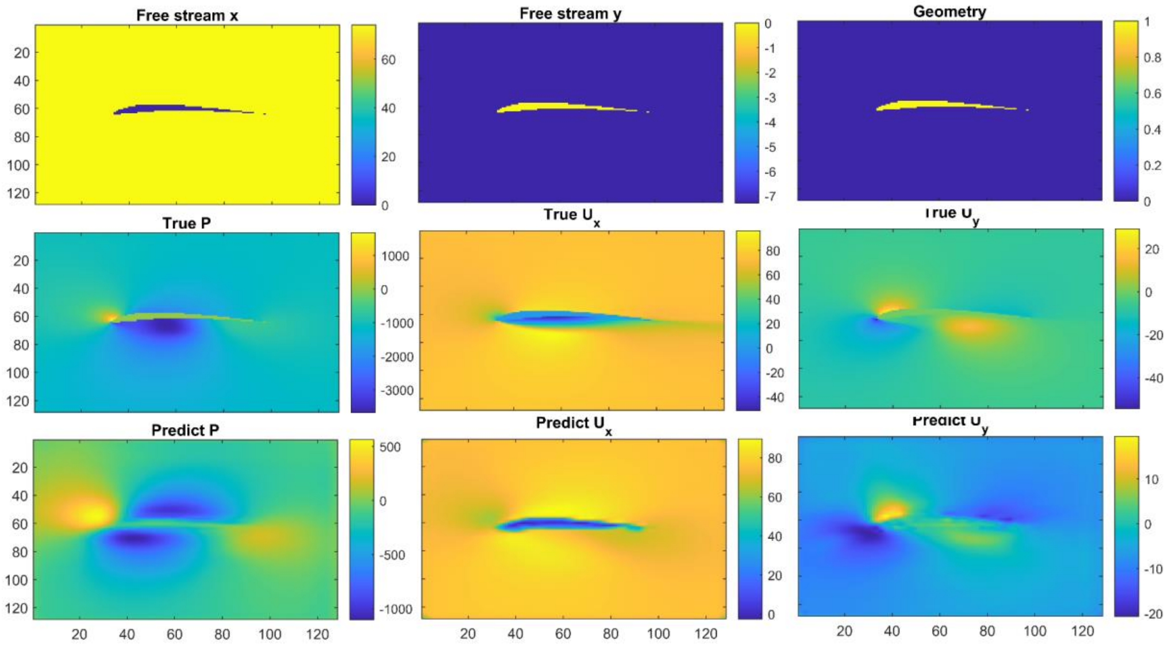

Figure 6. Changes in error over test data. The unit of velocities component is *m/s* and the unit of the pressure is *atm*.

Figure 6 presents the results for a very thin airfoil at high velocity and zero angle of attack. The thin airfoil is sufficiently good to delay the separation flow at low air density and low pressure, which should often be applied for unmanned aerial vehicles in the stratosphere. The thin airfoil could be a big problem for the network due to the low resolution of the data. However, as can be seen, the pressure and velocity field around the model can be predicted well from the current network. For the pressure at the upper surface, some inaccurate results occur, which is probably from insufficient data for the training process. The findings regarding the airfoil indicate that the average error in pressure and velocity fields is below 3%. While this error rate is deemed acceptable, there are inquiries regarding the potential for further reduction. Clearly, augmenting the volume of training and testing data holds promise for minimizing calculation errors. However, such an augmentation demands substantial computer resources, posing a significant challenge, particularly within the context of Vietnam's conditions. Additionally, enhancing the artificial neural network architecture could aid in error reduction. Nevertheless, in our investigations, employing 4 or 5 layers of the U-Net network did not yield a substantial decrease in calculation error despite the increased network parameters. Exploring alternative types of artificial neural networks, such as Flownet or novel architectures, may offer avenues for reducing calculation errors. Another area of concern is the precision of pressure distribution on the model surface when employing such artificial neural networks. While certain other networks may enhance the accuracy of pressure field distribution prediction, the current state of artificial neural networks fails to fully encapsulate the physical constraints of flow, such as the continuity of velocity and pressure fields. Therefore, integrating physical constraints into the input data could potentially enhance the model's accuracy. These specific inquiries will be addressed in detail in our forthcoming research endeavors

3.4. Application of the Network in determining the flow fields around the blunt-base model

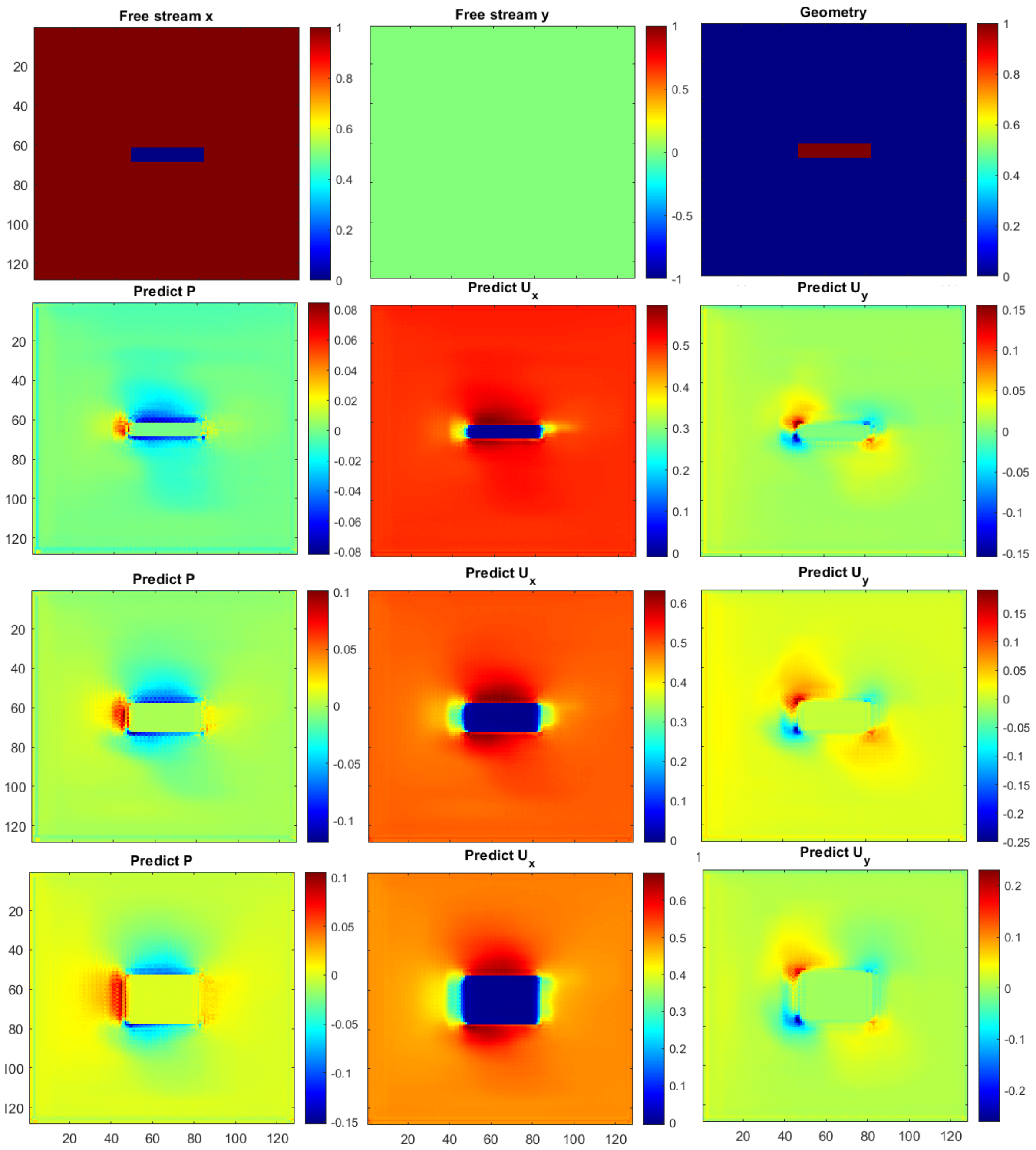

Figure 7. Flow around the blunt body by the current convolutional neural network. The *x* and *y* axes present the pixel. Flow is from left to right. The unit of velocities component is *m/s* and the unit of the pressure is *atm*.

In this section, flow around the blunt body model is extracted to show the ability of the network to determine the flow. Here, the blunt body with different aspect ratios from 1.7 to 8.5 was investigated. We divided the data into ten cases with a fixed length. The large length of 8.5 is similar to a model in the previous study by Tran et al. [11], [12]. The short length of the model was studied by Nonomura et al. [13]. The velocity inlet of two components is at (1, 0) m/s. As can be seen in Figure 7, the high pressure at the nose of the models can be predicted well from the networks. Similar to Figure 2, here, the *x* and *y* axes present the pixel number and the flow is from the left to right. Since this study focuses on image analysis and the axe presents the pixel number, we do not present them in the figure. As the fluid approaches the nose of the model, the pressure is high and velocity becomes low. Behind the base, a low-velocity region is also formed. The network allows us to predict the pressure and velocity fields well and the basic concept of flow physics can be understood. However, for higher accuracy, the training process should include the blunt body. Additionally, a simulation should be conducted to obtain the training data. It is one of our further tasks.

4. CONCLUSION

In this research, a convolutional neural network (CNN) is developed to replicate the flow around an airplane wing model. The training dataset is generated by solving the Navier-Stokes equations using the Reynolds-Averaged Navier-Stokes (RANS) method. Despite the simplicity of the training model, the neural network contains a large number of hidden parameters. The training results indicate a good fit with a small error between the training data and the simulation outcomes. Consequently, the findings from this study can be utilized to predict the primary characteristics of the pressure and velocity fields around the model, aiding in the optimization of the airplane wing shape under specific operating conditions. The network is further employed to predict the pressure and velocity around a blunt-base body with different aspect ratios, yielding highly accurate results. Nevertheless, the model requires further refinement. Moreover, enabling the machine learning model to comprehend the physical phenomena of the flow remains a complex challenge that future research needs to address.

The other problem could be three-dimensional (3D) simulation and training. Although the generation of data for training of the 3D simulation is not complicated and can be done by us, the large size of the data can be a problem for the training process. As we can see from Figure 1, the matrix of the training data should have five dimensions for training 3D simulation. It should be a big problem for training using personal computer. However, since the principle of the training remains the same, this training process can be done. It is an important task of our further study.

ACKNOWLEDGMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 107.03-2021.52.

REFERENCES

[1]. M. Drela, XFOIL: An analysis and design system for low Reynolds number airfoils, in Low Reynolds Number Aerodynamics, in Proceedings of the Conference Notre Dame, Indiana, USA, 5–7 (1989) 1–12. https://doi.org/10.1007/978-3-642-84010-4_1

[2]. T.H. Tran, D.A. Le, T.M. Nguyen, C.T. Dao, V.Q. Duong, Comparison of numerical and experimental methods in determining boundary layer of axisymmetric model, in International Conference on Advanced Mechanical Engineering, Automation and Sustainable Development, (2022) 297–302. https://doi.org/10.1007/978-3-030-99666-6_45

[3]. D.Q. Nguyen, T.H. Tran, D.A. Le, N.T. Hieu, V.K. Pham, T.S. Ha, Drag reduction for axisymmetric boattail model by longitudinal groove cavity under low-speed conditions, Journal of Mechanical Science and Technology, 38 (8) (2024) 4209–4220. http://doi.org/10.1007/s12206-024-0718-4

[4]. G. Du, X. Cao, J. Liang, X. Chen, Y. Zhan, Medical image segmentation based on u-net: A

review, Journal of Imaging Science Technology, 64 (2020). http://doi.org/10.2352/J.ImagingSci.Technol.2020.64.2.020508

[5]. H.T. Tran, K.V. Pham, A.D. Le, C.T. Dinh, Frequency characteristics of axisymmetric conical boattail models with different slant angles, Physics of Fluids, 35 (2023) 095113. https://doi.org/10.1063/5.0160053

[6]. N. Thuerey, K. Weibenow, L. Prantl, X. Hu, Deep learning methods for Reynolds-averaged Navier–Stokes simulations of airfoil flows, AIAA Journal, 58 (2020) 25–36. http://doi.org/ 10.2514/1.j058291

[7]. N. Siddique, S. Paheding, C.P. Elkin, V. Devabhaktuni, U-net and its variants for medical image segmentation: A review of theory and applications. IEEE Access, 9 (2021) 82031-82057. http://doi.org/10.1109/ACCESS.2021.3086020

[8]. X. Wu, D. Hong, J. Chanussot, UIU-Net: U-Net in U-Net for infrared small object detection. IEEE Transactions on Image Processing, 32 (2022) 364-376. http://doi.org/10.1109/TIP.2022.3228497

[9]. P. Spalart, S. Allmaras, A one-equation turbulence model for aerodynamic flows. 30th aerospace sciences meeting and exhibit, 439 (1992). https://doi.org/10.2514/6.1992-439

[10].C. Kostić, Review of the Spalart-Allmaras turbulence model and its modifications to three-dimensional supersonic configurations, Scientific Technical Review, 65 (2015) 43-49.

[11]. K.V. Pham, H.T. Tran, D.T. Nguyen, A.D. Le, Q.D. Nguyen, D.T. Pham, Delayed detached eddy simulation for wake flow analysis of axisymmetric boattail models under low-speed conditions, Physics of Fluids, 36 (2024) 035159. https://doi.org/10.1063/5.0188363

[12]. T.H. Tran, The effect of boattail angles on the near-wake structure of axisymmetric afterbody models at low-speed condition, International Journal of Aerospace Engineering, (2020) 1-14. https://doi.org/10.1155/2020/7580174

[13]. T. Nonomura, K. Sato, K. Fukata, H. Nagaike, H. Okuizumi, Y. Konishi, H. Sawada, Effect of fineness ratios of 0.75–2.0 on aerodynamic drag of freestream-aligned circular cylinders measured using a magnetic suspension and balance system, Experiments in Fluids, 59 (2018) 1-12. https://doi.org/10.1007/s00348-018-2531-2