An Airfoil Aerodynamic Parameters Calculation Method Based on Convolutional Neural Network

Haolin Liu
Mechanical Engineering
Carnegie Mellon University
5000 Forbes Ave
haolinl@andrew.cmu.edu

Zi Li
Mechanical Engineering
Carnegie Mellon University
5000 Forbes Ave
zili@andrew.cmu.edu

Felix Lu
Mechanical Engineering
Carnegie Mellon University
5000 Forbes Ave
flu2@andrew.cmu.edu

Abstract

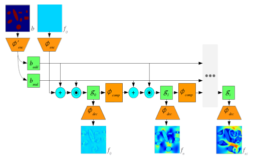
The method of convolutional neural network (CNN) is capable of image processing, which is widely used nowadays for aerodynamic meta-modeling task. Based on the dataset with adequate quantity, the aerodynamic property of airfoil can be predicted by using CNN. The primary objective of this paper is using CNN to predict the lift-to-drag ratio of airfoils with different angles of attack (α). The method of computational fluid dynamics (CFD) is also conducted in this paper to calculate the lift-to-drag ratio of all the airfoils. Basically, the efficiency and accuracy of these two methods are compared and discussed in this paper, and it is shown that the method of CNN can maintain a relatively competitive level of accuracy and has far better efficiency than analytical method and CFD method.

KEYWORDS: Convolutional Neural Network (CNN), Lift-to-drat Ratio, CFD, Airfoil.

1. Introduction

In the field of aerodynamics, most of the problems are traditionally handled by solving the corresponding partial differential equations (PDE). However, some problems, like flow fields prediction, are usually high dimensional, highly non-linear and multi-scale, which can be truly difficult to find an analytical solution or find ways to get a totally reasonable explanation. In a common situation, these hard-to-solve problems are handled by using numerical methods, which can help get numerical solutions and make an approximate to analytical solution. Nonetheless, numerical methods are usually time-consuming and highly possible to be diverged during the process of calculation.

The method of machine learning is widely used nowadays in different fields to solve problems of all kinds. With the development of computer science and the growing size of dataset, many efficient ways of calculation have merged and considerably amount of problems, including problems related to aerodynamics, has been addressed by using specific machine learning algorithms [4, 1]. The method of multi-layer perceptron (MLP) are commonly used to address the problems under framework of classification and regression, which is also effective in the area of aerodynamics. *Rai et al* implements MLP to achieve two-dimensional aerodynamic design, which should be a really hard task if worked in traditional ways [5]. Using convolutional neural network (CNN) recently is another common way to extract features of images, which makes contributions in lots of academic areas. *Guo et al* uses CNN to succesfully approximate the properties of laminar flow and make several predictions of it[3].


(a) Convolutional neural network model

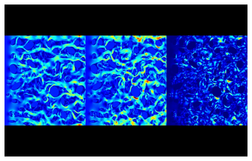

(b) Generated prediction image

Figure 1: Method of CNN

One of the most important merits of the machine learning method is its high working efficiency, therefore many engineering problems, especially those related to the numerical calculating (which could be extremely time consuming), can be solved in a relatively shorter time when certain and proper machine learning methods are implemented. With the development of machine learning itself, difference in calculating efficiency between different machine learning methods also turns out gradually and become more and more important. Rai et al implement different neural networks, including MLP and CNN, to achieve designment of turbomachinery airfoils simultaneously, and they find out

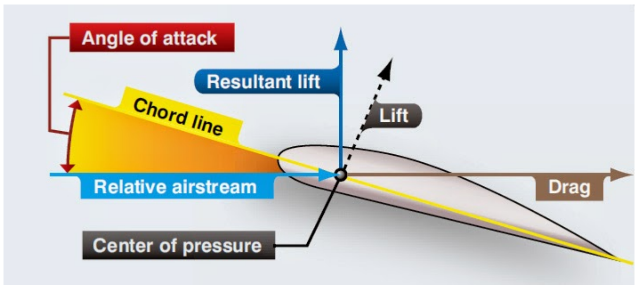


Figure 2: The lift-to-drag ratio of an airfoil ($\frac{c_L}{c_d}$)

that the method of CNN could be much faster than the method of MLP [5]. Since the input of CNN is generally figures and images, more effective information can thus be contained, and it is reasonable for CNN to achieve higher information exchanging.

With respect to an airfoil, which should be the cross-section of the wing of airplane, one of the most important tasks is to get its aerodynamic properties. One of the most crucial parameters of an airfoil is its lift-to-drag ratio ($\frac{c_L}{c_d}$). The $\frac{c_L}{c_d}$ basically refers to the ratio of resultant lift force over drag, which is shown in Figure 2. The $\frac{c_L}{c_d}$ represents the working efficiency of an airfoil and is extremely valuable to airfoils optimization. To calculate it, the vortex panel method is frequently used to get the numerical solution. But with its low calculating efficiency, in many cases this method is obviously time consuming. Therefore, based on the drawbacks of numerical methods and the advantages of CNN, the method of CNN is implemented in this paper to map airfoil shapes to the corresponding ($\frac{c_L}{c_d}$), and the angle of attack (α), which decides the position of airfoil, is considered in the work.

2. Related Work

Implementing neural network algorithm to the area of airfoil property prediction is not a new topic among the field of aerodynamic design and calculation. One and a half decade ago, Suresh et al has used recurrent neural network (RNN) to predict the lift coefficient of the airfoil [6]. In their work, method of RNN is taken as a effective way to solve highly non-linear problems; based on this high level idea, considerably good are obtained and the model built in their paper takes time and space complexity to some extent. Nonetheless, since the method of CNN is not developed then, the method that mapping input to the output of lift coefficient takes some unnecessary complexity and relatively more time is used to train the RNN model.

In recent years, more and more aerodynamic researchers pay attention to the implementing of CNN on problems related to airfoil design and calculation. Yilmaz and German implement CNN on predicting the pressure distribution along the contour of the airfoil [2]. Under the framework of classification, the pressure coefficient along the airfoil is discretized and predicted. A number of accuracy is calculated in that paper and it shows that an accuracy of over 80 percents is achieved. But the structure of CNN used in their work contains basically two convolutional layers, the features extracted form which could be too less and may be not effective to the learning effect of the model; moreover, the input used only shows the contour of the airfoil, which may not contain enough information and may cause relative inaccuracy to the result they generate.

Based on these drawbacks, Yao et al successfully implement CNN on calculating lift coefficient of airfoil [7]. The α, Reynolds number (Re) as well as Mach number (Ma) are considered in their model and the input image of fully-filled airfoil takes a resolution of 49×49. With an input dataset of such size, accurate prediction model is obtained and impressive predicting results are gotten from this model. But because the resolution of the input figure is relatively small, the increment of α can only be $2°$ so that there can appear difference between figures of airfoils with different angles of attack. Also, the model only takes the lift coefficient into account, the effect of drag force or, basically $\frac{c_L}{c_d}$, is not considered. In our paper, we increase the resolution to 128×128 to contain more information and make the $1°\alpha$ increment possible, and the lift-to-drag ratio is also studied.

3. Data Processing

3.1. Image Generation

To find the relationship between lift-to-drag ratio and airfoil shape through neural network, the first and critical step is preparing input data with proper mathematical representation for training the neural network.

A dataset with quantity adequacy and type diversity is required to make the neural model robust to be applied on different airfoils. With this purpose we choose UIUC Airfoil Data Site, which contains coordinates of nearly 1600 mainstream airfoils from NACA 4-digit series to Selig series, as our raw data.

There are several ways of processing the raw data. The first way is to input the coordinate pairs into the network. Thus input data is a 2N-element vector, in which N is the number of coordinate pairs. However, this seemingly straightforward method will result in problem when deciding the number of neurons in the input layer since different airfoil sample have slightly different number of coordinate pairs. To address this nonuniformity problem in data size, we transform the raw coordinate pairs into greyscale images with the same size. For each airfoil sample, the raw (2, N) coordinate matrix is firstly plotted as contour image of 128 pixels by 128 pixels. Notice that image size is set by (128,

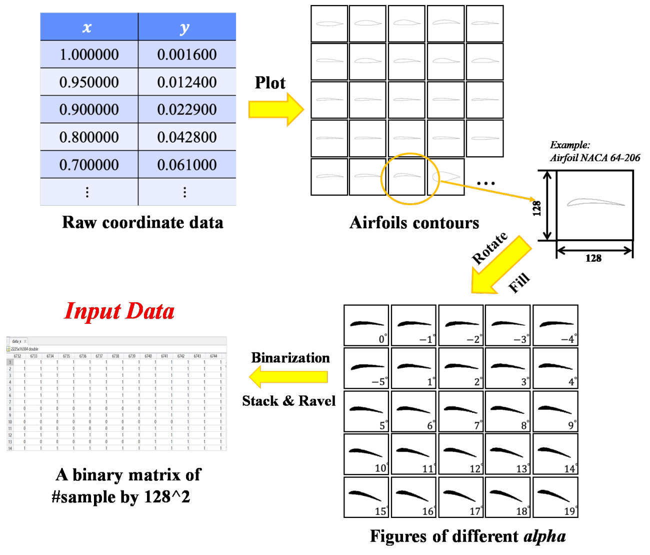


Figure 3: Image generating & Input dataset creating

128) instead of the normally smaller size, such as (28, 28) in MNIST dataset or (32, 32) in NIST36 dataset, since aerodynamic performance of airfoils is very sensitive to subtle changes of its shape and it is of critical importance to maintain a decent resolution to keep prediction accuracy. However, it will be computational expensive and unnecessary to make the contour size with resolution as high as (256, 256) in ImageNet dataset since there are not many subtle details information in the airfoil. Weighting over both accuracy and computation, a image size of (128, 128) will be the best fit for the application in airfoil data.

After plotting the contour of airfoils, the next step is to fulfill the contour to bring more valuable learning materials to the network instead of bringing in only few useful contour pixels within large useless blank background pixels. We apply opening method, which is an erosion-followed-by-dilation morphology manipulation in computer vision, to fill in the blank space within the contour.

To take into account the effect of angles of attack have on the airfoil aerodynamic performance, the filled-in contour image (which is based on initial zero attack angle) is rotated with respect to the chord line from -5 to +19 with an interval of 1 . Thus for each raw airfoil sample, we produce 25 images. After binarization and flattening, we stack all the sample data together and form a (N, 162384) matrix between 0 and 1, and this will be the input data of the neural network model.

3.2. Ground Truth Calculation

Since the airfoil is only a 2D shape of the cross section of the airplane's wing, it will be hard to get results from

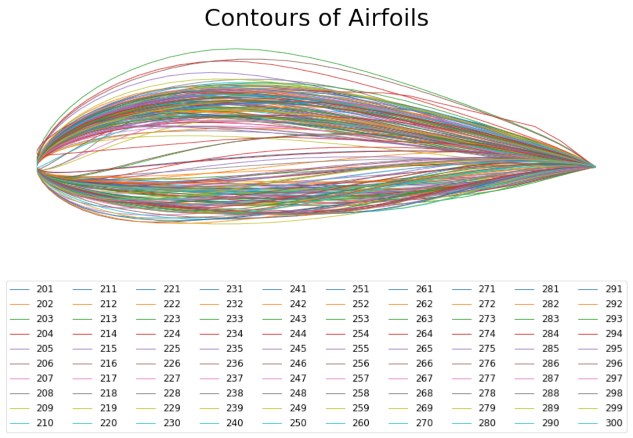


Figure 4: Contours of Airfoils (i.e. No.201 - No.300)

experimental ways. To quickly get the ground truths of the lift-to-drag ratio, certain CFD software is used to calculate the corresponding parameters and save the results for future use. In this paper, a software named *xflr5* is implemented to get the required results.

3.2.1 Import Airfoil Figures

Firstly, to use the data downloaded from UIUC Airfoil Dataset, all the data files should be imported into the software and generate corresponding figures of contours. Certain number of panels are set to ensure the convergence of calculation. The image of contours is shown exactly as Figure 4.

With these contours, different results of $\frac{c_L}{c_d}$ can be obtained and the influence of α is considered automatically. Each converged result of $\frac{c_L}{c_d}$ and its corresponding α will be recorded and exported and regarded as one sample.

3.2.2 Calculate Lift-to-drag Ratio

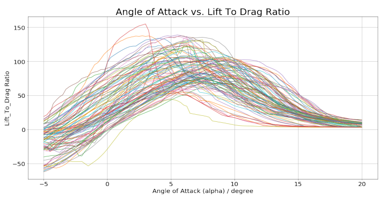
By choosing the numerical method, the results of $\frac{c_L}{c_d}$ can be obtained from the software. An corresponding figure of different group of the relationship between airfoil's α and $\frac{c_L}{c_d}$ can be exported, which can generally be shwon as Figure 5.

4. Methods

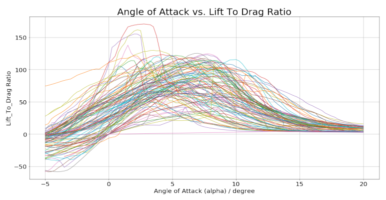
4.1. Theoretical Method

Traditionally, the problems of aerodynamics can possibly be solved in analytical ways. In the airfoil calculation area, the partial differential equation of *Navier-Stokes Equation* could be the main part of analytical method.

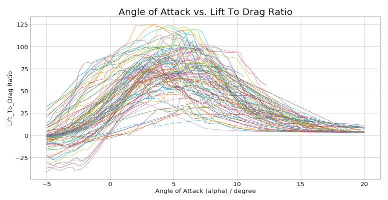
$$\begin{cases} \rho\left(\frac{\partial v}{\partial t} + (v \cdot \nabla)v\right) = -\nabla p + \mu\nabla^2 v + f, \\ \nabla \cdot v = 0, \end{cases} \tag{1}$$



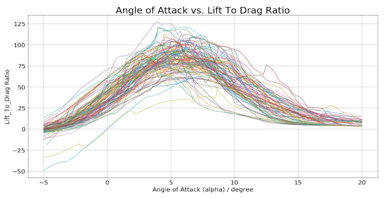
(a) 201-300



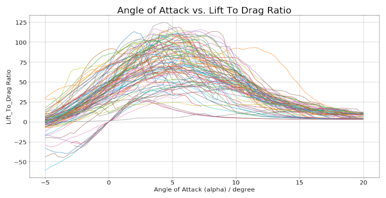
(b) 301-400



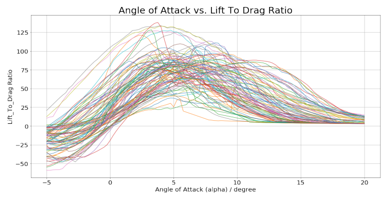
(c) 501-600



(d) 601-700



(e) 701-800



(f) 1001-1100

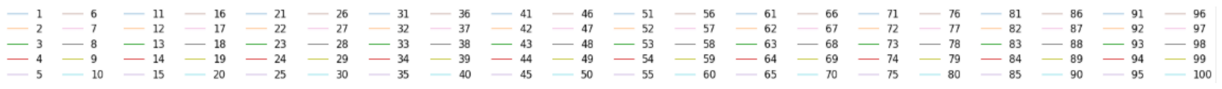

(g) legend of figures

Figure 5: Plot of relations α vs. $\frac{c_L}{c_d}$

Equation (1) is the special form of the *Navier-Stokes Equation* based on the assumption of viscous incompressible flow. Considering the situation of convective acceleration, the velocity of the flow may not merely the function of time. In that case, the equation can further be written as:

$$\begin{cases} \rho\left(\frac{\partial v_x}{\partial t}+v_x\frac{\partial v_x}{\partial x}+v_y\frac{\partial v_x}{\partial y}+v_z\frac{\partial v_x}{\partial z}\right) \\ =\mu\left[\frac{\partial^2 v_x}{\partial x^2}+\frac{\partial^2 v_x}{\partial y^2}+\frac{\partial^2 v_x}{\partial z^2}\right]-\frac{\partial p}{\partial x}+\rho g_x, \\ \\ \rho\left(\frac{\partial v_y}{\partial t}+v_x\frac{\partial v_y}{\partial x}+v_y\frac{\partial v_y}{\partial y}+v_z\frac{\partial v_y}{\partial z}\right) \\ =\mu\left[\frac{\partial^2 v_y}{\partial x^2}+\frac{\partial^2 v_y}{\partial y^2}+\frac{\partial^2 v_y}{\partial z^2}\right]-\frac{\partial p}{\partial y}+\rho g_y, \\ \\ \rho\left(\frac{\partial v_z}{\partial t}+v_x\frac{\partial v_z}{\partial x}+v_y\frac{\partial v_z}{\partial y}+v_z\frac{\partial v_z}{\partial z}\right) \\ =\mu\left[\frac{\partial^2 v_z}{\partial x^2}+\frac{\partial^2 v_z}{\partial y^2}+\frac{\partial^2 v_z}{\partial z^2}\right]-\frac{\partial p}{\partial z}+\rho g_z, \end{cases} \tag{2}$$

where

- ρ: Density of the flow
- v: Velocity of the flow
- μ: Viscosity

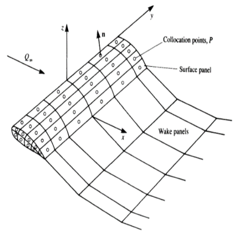
And considering the continuity:

$$\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}=0 \tag{3}$$

The above equations is the simplified form of *Navier-Stokes Equation* based on the assumption of viscous incompressible flow. These equations are hardly to get analytical solutions because the physics under this equation is truly hard to figure out. Normally, getting an approximate solution of this equation is much more feasible, but the process is still time-consuming and extremely inconvenient.

4.2. Computational Fluid Dynamics (CFD Method)

Due to the inconvenience of getting analytical solution, numerical method is therefore emerged. Numerical methods of solving problems related to aerodynamics include vortex panel method, vortex-lattice method, etc. Calculating parameters of airfoil can be solved by vortex panel method, which is shown as Figure 6.



(a) Panels of airfoil and flow

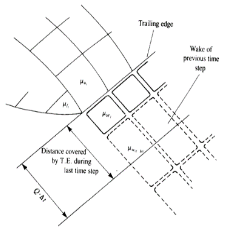

Trailing edge
Wake of previous time step
Distance covered by T.E. during last time step
(b) Consistency on the trailing edge

Figure 6: Vortex Panel Method

Vortex panel method basically focus on getting numerical result that can approximate the analytical solution. This mainly refers to solving the following series of equations:

$$\begin{cases} U = U_\infty + u, \\ \Phi^* = \Phi_\infty + \Phi, \\ \nabla^2 \Phi = 0, \end{cases} \tag{4}$$

where the equation of Φ^* should be:

$$\begin{aligned} \Phi^* = \frac{1}{4\pi} \int_{S_B} \left[\frac{1}{r} \nabla(\Phi - \Phi_i) - (\Phi - \Phi_i) \nabla \frac{1}{r} \right] ndS \\ + \frac{1}{4\pi} \int_{S_W + S_\infty} \left[\frac{1}{r} \nabla \Phi - \Phi \nabla \frac{1}{r} \right] ndS \end{aligned} \tag{5}$$

Further derivation should be:

$$\begin{aligned} \Phi^* = & \frac{1}{4\pi} \iint_{S_B + S_W} \mu n \cdot \nabla \left(\frac{1}{r} \right) dS - \\ & \frac{1}{4\pi} \iint_{S_B} \sigma \left(\frac{1}{r} \right) dS + \Phi_\infty \end{aligned} \tag{6}$$

Further derivation should be:

$$C_p = \frac{p - p_{ref}}{pv_{ref}^2/2} = 1 - \left(\frac{Q}{v_{ref}} \right)^2 - \frac{2}{{v_{ref}}^2} \frac{\partial \Phi}{\partial t} \tag{7}$$

where theC_p refers to the pressure coefficient; getting the distribution of C_p along the airfoil can help get the final result of the $\frac{c_L}{c_d}$.

Although the numerical method makes the problem solving feasible in certain situations, it is usually time-consuming and thus may not have a generally high level of calculating efficiency. It becomes a influential drawback and therefore in many cases the result of airfoil's property cannot be gotten immediately after the boundary condition and flow's properties are obtained.

4.3. Convolution Neural Network

For the regression problem in this case, in which the output is continuous, neural network model acts like a black box mapping airfoil images to their respective lift-to-drag ratio values. It has been demonstrated mathematically that any network with more than 2 layers can map any non-linear functions. However, the prediction result will be distinguishably different in the real application due to the quality of data, parameter tuning and other factors. Hence selecting a proper and powerful network model is of critical importance.

Traditional Multi-Layer Perceptron (MLP) is not appropriate for processing airfoil images input since the number of parameters becomes extremely high for (128, 128) resolution images and such fully-connected neural network will easily fall into over-fitting. Compared with MLP, CNN has demonstrated strong capability in image recognition

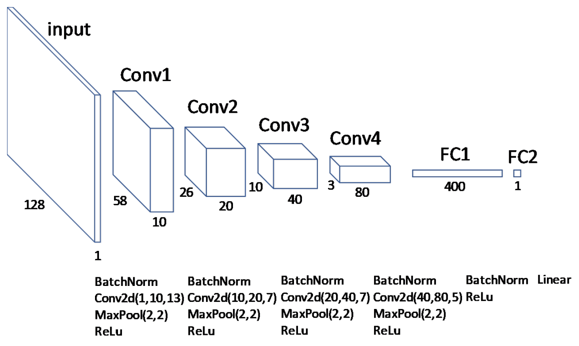

input
Conv1
Conv2
Conv3
Conv4
FC1
FC2
128
1
58
10
26
20
10
40
3
80
400
1
BatchNorm Conv2d(1,10,13) MaxPool(2,2) ReLu
BatchNorm Conv2d(10,20,7) MaxPool(2,2) ReLu
BatchNorm Conv2d(20,40,7) MaxPool(2,2) ReLu
BatchNorm Conv2d(40,80,5) MaxPool(2,2) ReLu
BatchNorm ReLu
Linear

Figure 7: Structure of CNN

and other vision applications[2]. CNNs feature extraction method by convolution operation also makes it detect local features better. It also has fewer parameters to learn because of parameters sharing mechanism. In this project, we choose CNN as our model

The CNN architecture is composed of two parts: Encoder, which extracts features, and Decoder, which learns the features and outputs regression results.

The Encoder contains 4 convolutional layers. In each convolutional layer, batch normalization, an optimization technique, is applied to shift and scale the data from last layer. [3]It has been shown that by batch normalizing neural networks can be trained faster and have higher accuracy. Then features are extracted by applying kernels filtering the image and obtained feature maps.

To reduce the amount of parameters we conduct max pooling here to reduce the size of the feature maps to their half. The last step is to pass the pooled result into ReLu activation function to increase the non-linearity of the network. The kernel size for each layer is reducing from 13 to 5 while the number is increasing from 10 to 80, so that the kernel can extract subtler and higher-level representation of the airfoil images.

The Decoder contains of 2 fully-connected layer. For regression problem, we choose only 1 neuron to output the final prediction result.

5. Experiment

A convolution neural network(CNN) is built to compute/predict the lift-to-drag ratio. To prove our approach is accurate and efficient, different metrics are used, such as mean squared error (MSE), confusion matrix and time-consumption table.

This paper took 1000 airfoil samples from 1550 airfoil sample datasets that generated previously, then split them into three parts: 70% training datasets, 20% validation datasets and 10% testing datasets. The purpose of validation datasets is to compare the training loss and validation loss during training time, so that we can fine-tune the parameters. The range of attack-angle is from -5 degree to +20 degree, the increment is 0.25 degree, which generates 100 images for each airfoil sample. For the final experiment, we chose 20578 images from 1000 airfoil samples, transformed them into a matrix which the size is 20578 by 16384. And the label size is also 20578 by 1.

All the training and testing stages in this paper have been completed by using the Pytorch package in GPU. Pytorch is an open source software library originally developed by the Facebook team for use in machine learning.

5.1. Fine-tuning Hyper-parameters

In the beginning, hyper-parameters are set as follows:

- batch size = 50
- learning rate = 1e-3
- epoch = 50
- dropout = 0.5 (for each layer)

We discovered that the learning rate is considerably large, which led to divergence of the loss or local minimum. So, we tried different learning rates, found out that learning rate of 5e-5 fits our model best. Then we noticed the accuracy is not high enough, which is related to the dropout layers we used. By getting rid of all the dropout layers other than the last FC layer, we yielded pretty good accuracy, and the value between train loss and validation loss seemed reasonable. Furthermore, we downsized the mini batch size to 32, to reach a higher accuracy. The final hyper-parameters are showed below:

- batch size = 32
- learning rate = 5e-5
- epoch = 200
- dropout = 0.5 (for layer FC1)

6. Results and Discussion

After data pre-processing and CNN model building, the training process (also the model's learning process) is conducted and several training results are obtained. The results mainly focus on three aspects: training and validation loss, confusion matrix, model's accuracy and model's efficiency.

6.1. Training and Validation Loss

Figure 8 shows the performance of our CNN model during 200 epochs. The MSE loss of training datasets and validation datasets both converged after 50 epochs. Regardless of the diversity in the number of adjustable parameters or learning capabilities, they show similar learning trajectories

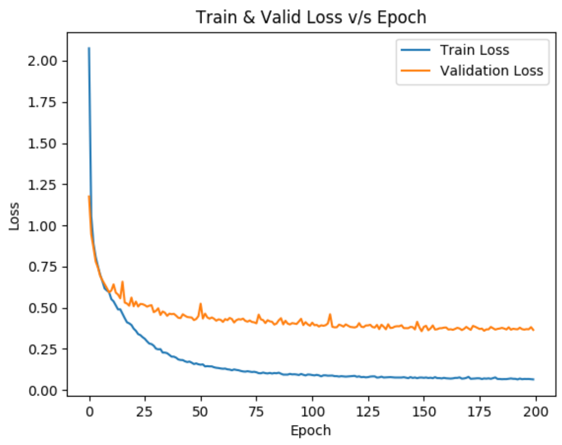
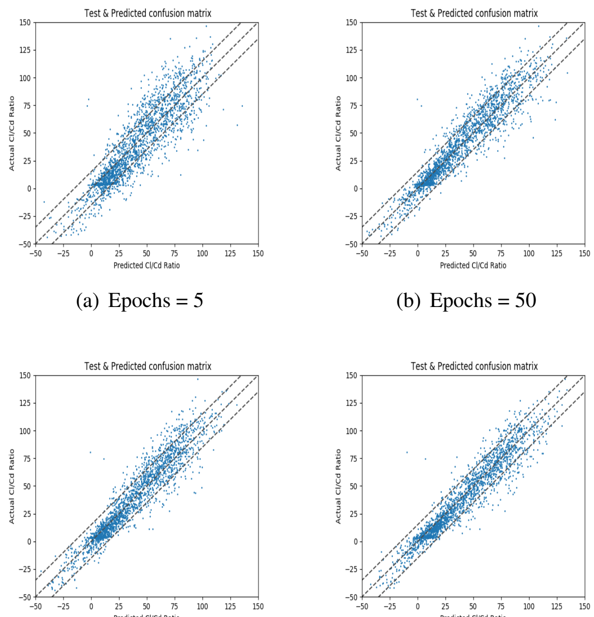


Figure 8: Training and Validation Loss(MSE)

resulting in the overall boundary of the obtainable accuracy for the lift-to-drag coefficient prediction. The training loss is 0.36484, and the validation loss is 0.06415 in the end.

6.2. Confusion Matrix



(a) Epochs = 5 (b) Epochs = 50 (c) Epochs = 100 (d) Epochs = 200

Figure 9: Confusion matrices with different training epochs

Figure 9 represents the deviation between the ground truth and prediction results from test datasets. The x-axis is prediction results, the y-axis is the ground truths. If the prediction output from our CNN model is accurate enough,

Epoch	Variance
5	114.075
10	94.098
30	73.335
50	59.085
100	54.955
150	52.819
200	48.174

Table 1: Variance of confusion matrix over epochs

the blue points will cluster along the diagonal line, meaning that predicted values are closer to actual values. There is a noticeable improvement from epoch 5 to epoch 200, which proves our CNN model is sufficient enough.

We also used the principle component analysis (PCA) to reconstruct the data in confusion matrix, reduce the dimension, compute the variance of confusion matrix. The variance is showed in the table 2 below.

6.3. Prediction and Ground Truth Comparison (De-normalized)

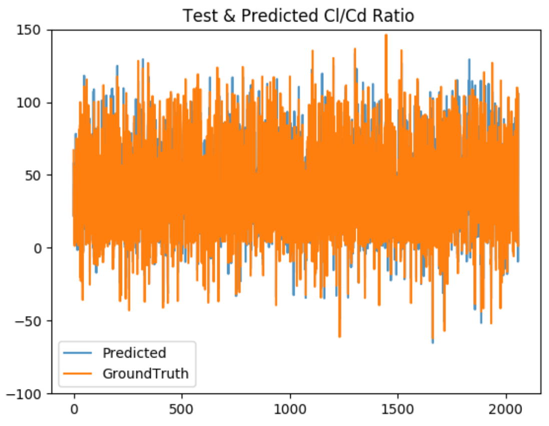


Figure 10: Deviation, Epoch = 50

Figure 10 shows the comparison between prediction and ground truth. Orange lines represent the ground truth C_L/C_D of each airfoil sample in one angle calculated by CFD. Blue lines represent our CNN model prediction results. X-axis is foil serial number, y-axis is lift-to-drag ratio.

Figure 11 gives us a closer look into one airfoil. We can see that two plots almost overlap, which means our prediction matches ground truth with high accuracy.

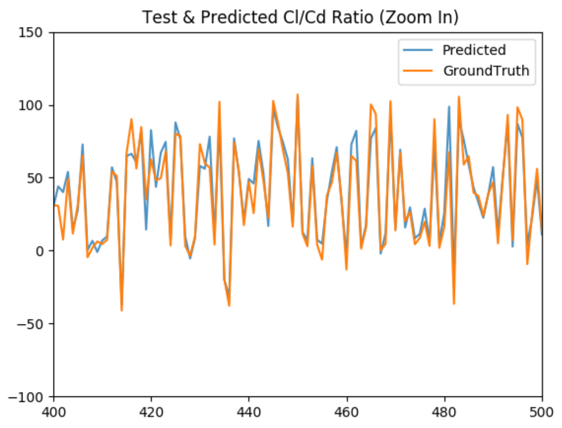


Figure 11: Prediction and Truth Comparison (zoom in)

NUM.	Time(s)	
	xflr5(s)	Well-trained CNN(s)
10	30.95	0.034
30	91.16	0.051
50	181.72	0.071
100	277.54	0.118
200	836.21	0.224
300	1897.35	0.308

Figure 12: time consumption table

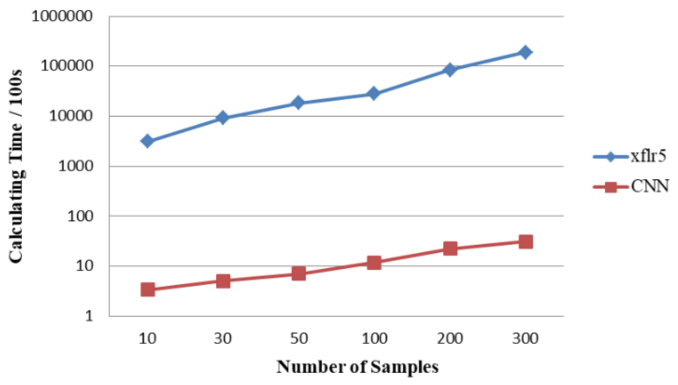


Figure 13: Time consumption

6.4. Efficiency

For CFD method, the solver will traverse all samples and compute multiple times until the results converge, so the time-complexity is $\mathcal{O}(n^2)$. In contrast, our CNN model after fine-tuning and training, will only traverse all samples once and output the results, which means the computing time increase linearly, thus, the time-complexity is $\mathcal{O}(1)$. When the size of testing samples is 300, the CNN method is 5000 times faster than CFD method.

7. Conclusion

With the results ontained and shown above, several following conclusions can be drawn:

- The calculating efficiency of CNN is 5,000 times higher than the calculating efficiency of CFD;

- The CNN we built can maintain a relative high level of accuracy. Accuracy of CNN will increase with the growth of epochs. The accuracy of CNN with 10 epochs is 62.68%, and the accuracy with 100 epochs is 83.09%;

- With this trained high-accuracy CNN, airfoil optimization can be achieved in the future;

- We revised the bad data in the UIUC Airfoil Dataset , and for each airfoil we generated filled airfoil figures for each increment of angle of attack, which can benefit researchers in the future.

In the future, based on the obtained trained CNN model, the work of optimization of airfoil can be conducted and further achieved.

Acknowledgement

This project and report cannot be finished without the help from others. The authors are very thankful to the teaching assistant who gave us many useful suggestions. We are also thankful to the instructor of this course, who also gave us some high level guidance and really cared about our project. Special thanking to the UIUC Airfoil Dataset and the CFD solver software xflr5, which are the most important dataset and tool of this project.

References

[1] T. B., K. Duraisamy, and J. Alonso. Application of supervised learning to quantify uncertainties in turbulence and combustion modeling. *AIAA Paper*, 259, 2013.

[2] Y. E. and G. B. A convolutional neural network approach to training predictors for airfoil performance. *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 3660, 2017.

[3] X. Guo, W. Li, and I. F. Convolutional neural networks for steady flow approximation. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM*, pages 481–490, 2016.

[4] Z. Z. J. and K. Duraisamy. Machine learning methods for data-driven turbulence modeling. *AIAA Aviation*, 2460, 2015.

[5] R. M. M. and M. N. K. Application of artificial neural networks to the design of turbomachinery airfoils. *Journal of Propulsion and Power*, 17(1):176–183, 2001.

[6] S. Suresh, S. O. , V. Mani, and T. G. Prakash. Lift coefficient prediction at high angle of attack using recurrent neural network. *Aerospace Science and Technology*, (7):595–602, 2003.

[7] Y. Zhang*, W. Sung, and D. Mavris. Application of convolutional neural network to predict airfoil lift coefficient. *AIAA SciTech Forum*, 2018.