An Airfoil Aerodynamic Parameters Calculation Method Based on Convolutional Neural Network

Haolin Liu, Zi Li and Felix Lu

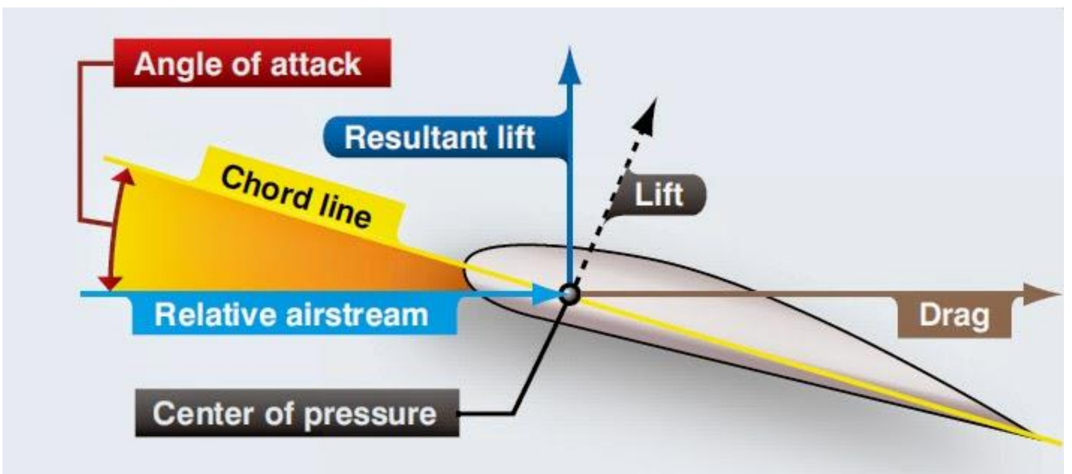
Carnegie Mellon University, Mechanical Engineering, 24-787B Artificial Intelligence and Machine Learning

Introduction

Lift-to-drag ratio is an important aerodynamics parameter of measuring the working efficiency of an airfoil. Its value depends on Angle of Attack (α), Reynolds number (Re, flow viscosity and stability) and Mach Number(Ma, flow speed). Aerodynamic scientists typically use two methods to calculate: one is numerical method (vortex panel method) and the other is analytical method (Navier-Stokes equation). But considering the computational expense, both of them are completely not efficient. To address this problem, we propose a cross-disciplinary method of fast and accurately calculating $\frac{C_L}{C_D}$ using convolutional neural network.

Background

Airfoil and its parameters

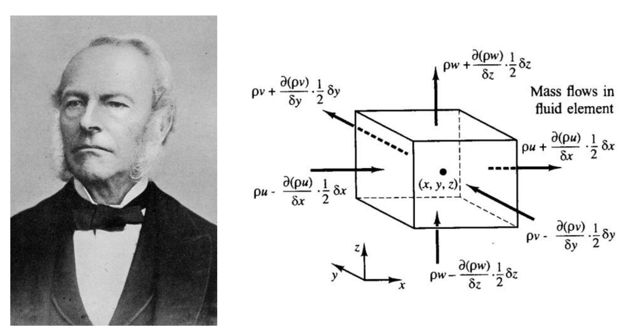


Airfoil: The cross section of airplane's wing

Analytical Method

Navier-Stokes Equation

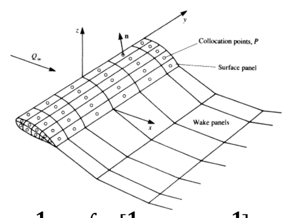
$$\begin{cases} \rho\left(\frac{\partial v}{\partial t} + (v \cdot \nabla) v\right) = -\nabla p + \mu \nabla^2 v + f \\ \nabla \cdot v = 0 \end{cases}$$



Numerical Method

Vortex Panel Method

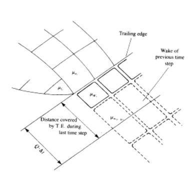
$$\begin{cases} U = U_\infty + u \\ \Phi^* = \Phi_\infty + \Phi \\ \nabla^2 \Phi = 0 \end{cases}$$



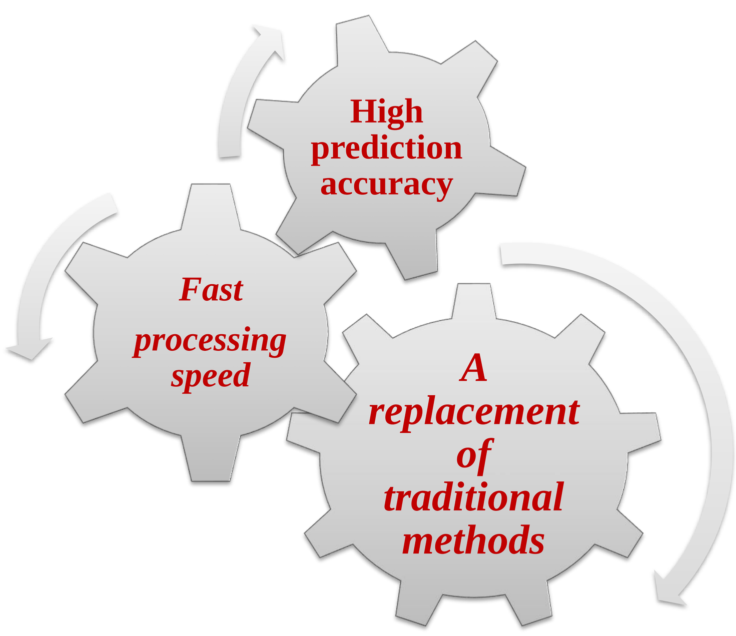
$$\Phi^* = \frac{1}{4\pi}\int_{S_B}\left[\frac{1}{r}\nabla(\Phi - \Phi_i) - (\Phi - \Phi_i)\nabla\frac{1}{r}\right] ndS + \frac{1}{4\pi}\int_{S_w + S_\infty}\left[\frac{1}{r}\nabla\Phi - \Phi\nabla\frac{1}{r}\right] ndS$$

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

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



Workflow



Model

A 6-layer Convolutional Neural Network to predict airfoil's aerodynamic parameters.

Dataset

UIUC airfoil dataset online, which contains coordinate data of 1,550+ airfoils.

https://m-selig.ae.illinois.edu/ads/coord_database.html

Tools

Pytorch (GPU version), Xflir5

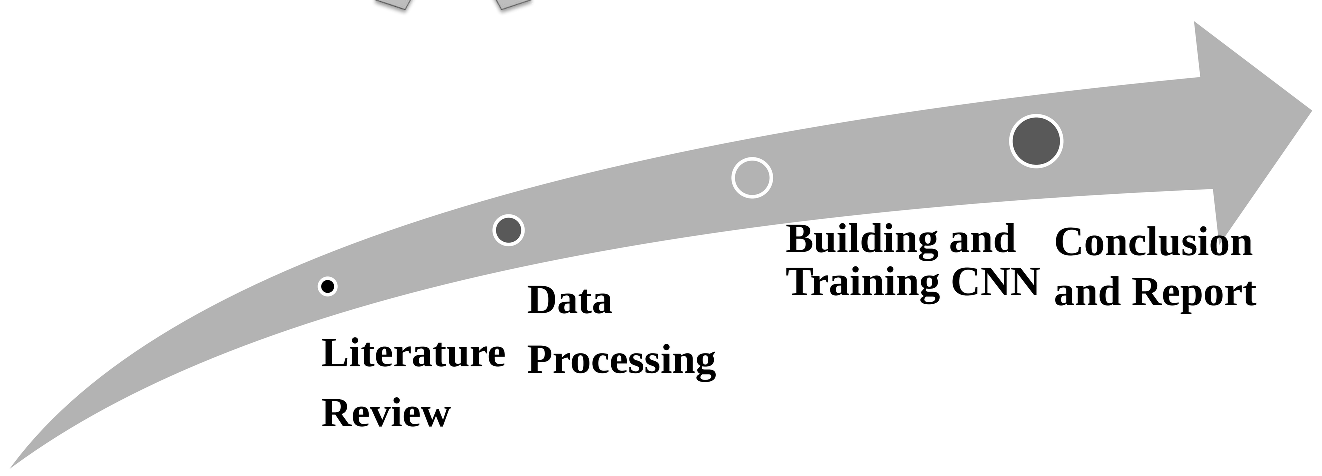
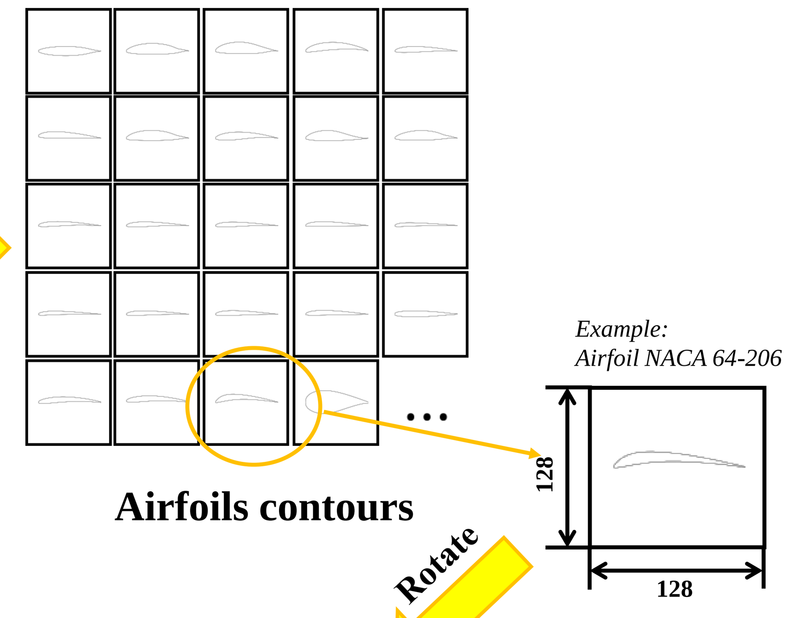


Image preprocessing

x	y
1.000000	0.001600
0.950000	0.012400
0.900000	0.022900
0.800000	0.042800
0.700000	0.061000
⋮	⋮

Raw coordinate data

Plot

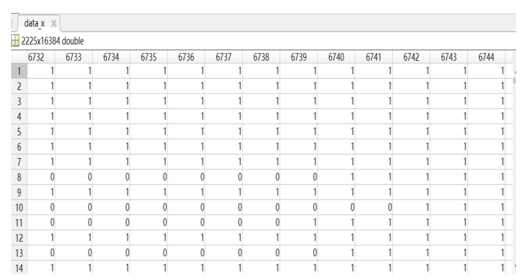


Airfoils contours

Rotate

Fill

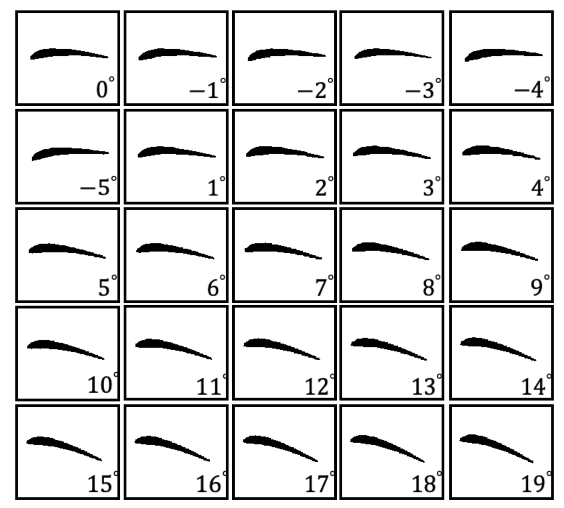
Input Data



A binary matrix of #sample by 128^2

Binarization

Stack & Ravel



Figures of different *alpha*

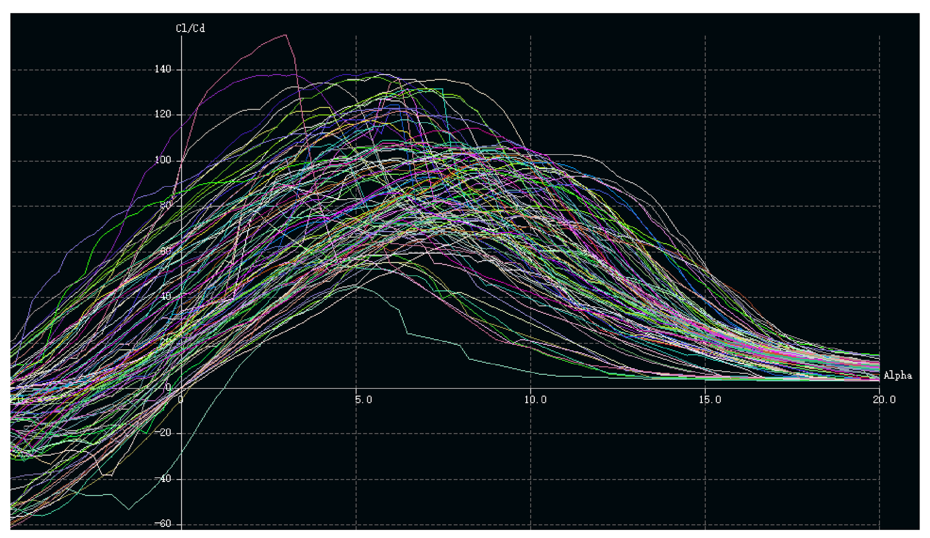
Ground Truth Calculation

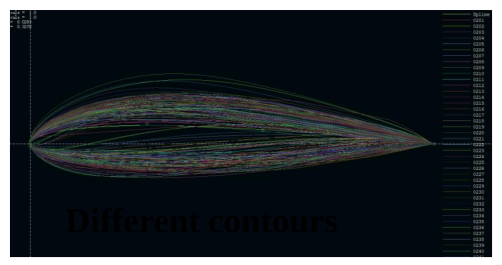
Raw Coordinates
Viscous incompressible flow
Reynolds number: 500,000
relatively low velocity flow
Mach number: 0

Import

xflr5

Visualize



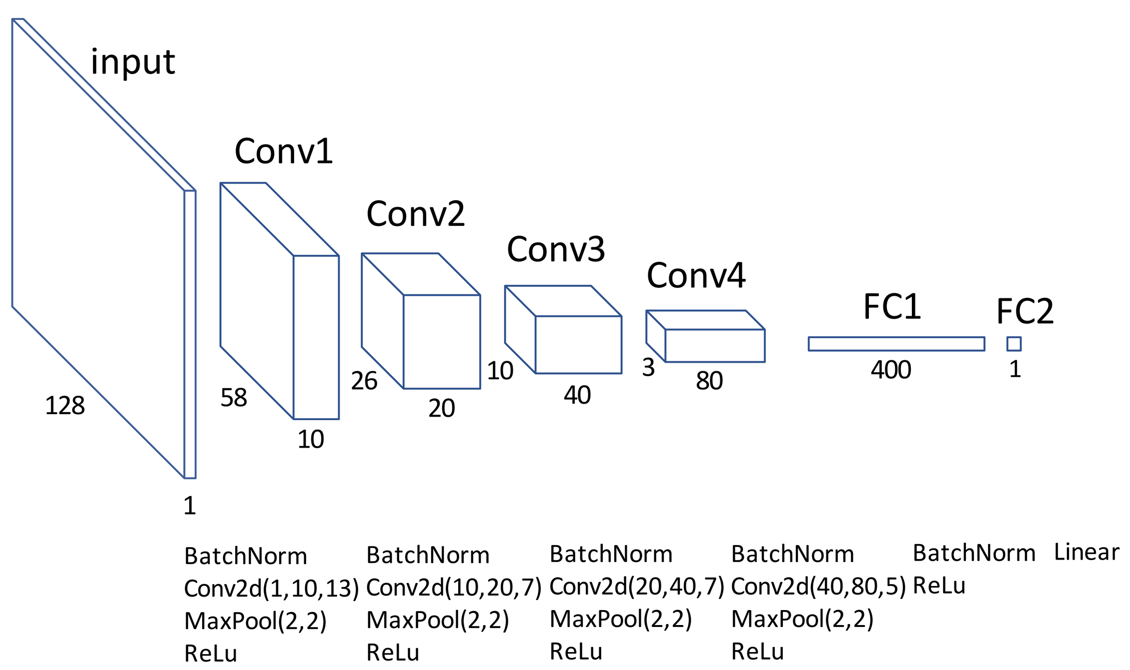


Batch CFD Calculation

One airfoil's Lift-to-drag ratio result w.r.t attack angles

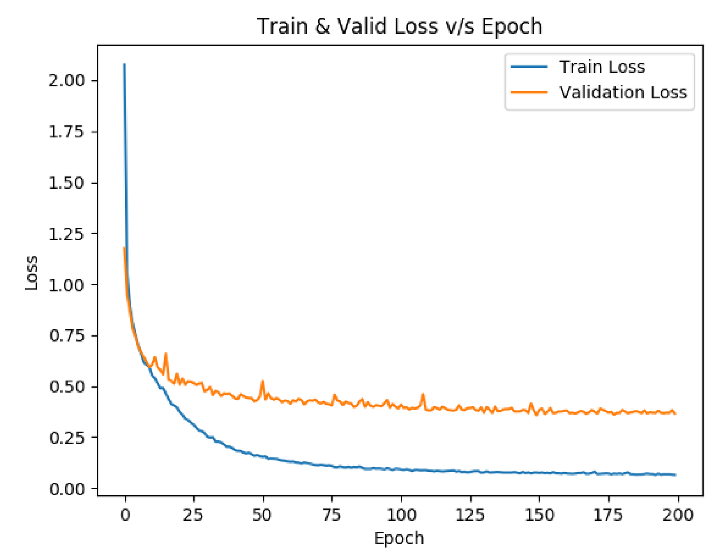
Output Data

CNN Model



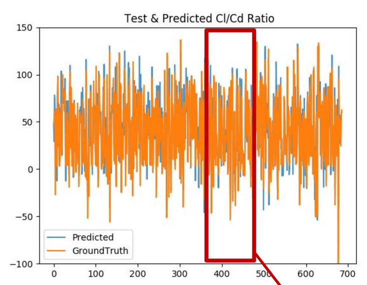
Results

Training and Validation Loss

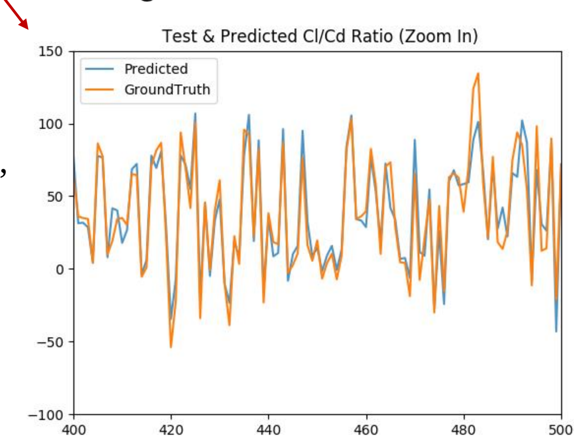


Validation MSE	**0.36484**
Train MSE	**0.06415**

Prediction and Truth Contrast

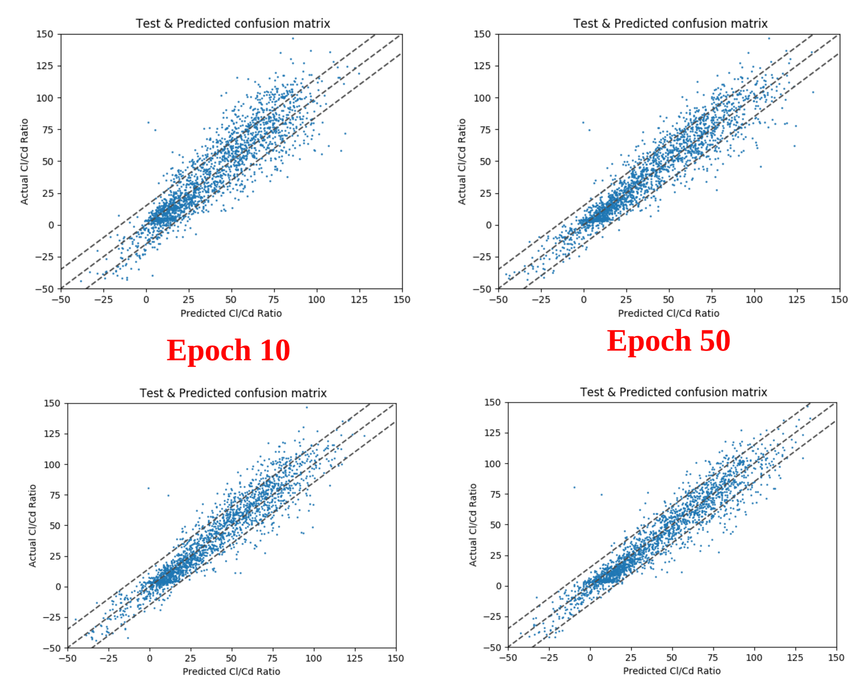


Orange lines represent the ground truth $\frac{C_L}{CD}$ of each foil sample calculated by CFD, blue lines represent our model prediction result. X-axis is foil serial number, y-axis is lift-to-drag ratio value.



We can see that two plots almost overlap, which means our prediction matches ground truth with high accuracy.

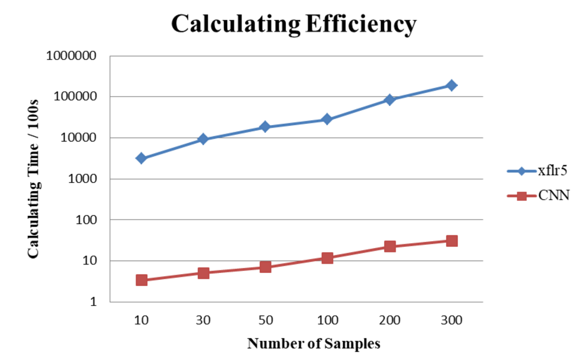
Deviation



Efficiency

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

Time Complexity
Xflr $\approx O(n^2)$
CNN $\approx O(1)$



CNN is 5000x faster than Xflr5.

Conclusions

1. The calculating efficiency of CNN is 5,000 times higher than the calculating efficiency of CFD.
2. 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%.
3. With this trained high-accuracy CNN, airfoil optimization can be achieved in the future.
4. 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.

References

[1] Müller, S., Milano, M., and Koumoutsakos, P., "Application of machine learning algorithms to flow modeling and optimization," Center for Turbulence Research Annual Research Briefs, Stanford University, Stanford, CA, 1999, pp. 169–178.
[2] Rai, M. M., and Madavan, N. K., "Application of artificial neural networks to the design of turbomachinery airfoils," Journal of Propulsion and Power, Vol. 17, No. 1, 2001.
[3] Emre Yılmaz and Brian J. German, "A Convolutional Neural Network Approach to Training Predictors for Airfoil Performance", Georgia Institute of Technology, Atlanta, GA, 2017.