Artificial Intelligence Based Method for Obtaining Aerodynamic Characteristics of Airfoils

Asif Mushtaq Ahmed Ansari[1], Hanish[2], Santosh Maganahally[3], Harish S.[4]

[1]Aeronautical Engineer, Amity Institute of Technology, Amity University, Noida, India
[2] Aeronautical Engineer, Amity Institute of Technology, Amity University, Noida, India
[3]Project Manager, Tata Technologies, Amity Institute of Technology, Amity University, Noida, India
[4]Consultant, Tata Technologies, Amity Institute of Technology, Amity University, Noida, India

---***---

Abstract - *In the discipline of aerodynamics, most issues are conventionally solved by solving the appropriate partial differential equations (PDE). However, some issues such as flow field prediction are often high dimensional, highly non-linear, and multi-scale making it extremely difficult to discover an analytical solution or provide a completely acceptable explanation. In most cases, these difficult to solve issues are treated utilising numerical methods, which can aid in obtaining numerical answers and producing an approximation to analytical solution. Nonetheless, numerical approaches are typically time-consuming and have a significant likelihood of diverging throughout the calculation process.*

Machine learning is currently widely employed in a variety of sectors to tackle challenges of all types. With the advancement of computer science and the increasing magnitude of datasets various efficient methods of computation have emerged.

This work involved obtaining aerodynamic characteristics of airfoils from numerical simulation tool JavaFoil and generation of Airfoil images based on coordinates obtained from UIUC Airfoil data repository. The images are later transformed to embed flow conditions (Reynolds Number, Mach Number). In reference to Neural Network, Pytorch software package was used and Python as programming language.

The developed convolutional neural network (CNN) models allow to choose any Mach number from 0-0.7, Reynold's number from 30000-1630000 and work on any airfoil. They can predict aerodynamic characteristics of airfoils faster compared to Computational Fluid Dynamics (CFD) method or any other numerical software. Hence, reducing time expenditure and computational cost associated with CFD analysis.

***Key Words*: Computational Fluid Dynamics (CFD), Artificial Intelligence (AI), Convolutional Neural Network (CNN), Dataset, Airfoil, Aerodynamic Characteristics.**

1.INTRODUCTION

Choosing the correct airfoil is an important step in the early stage of any aerial vehicle design since its shape has a direct impact on the overall aerodynamic characteristics of the aircraft or rotorcraft. Aerodynamic characteristics in addition to providing a measure of performance are used to create additional subsystems such as a flight control system and to anticipate complex dynamic phenomena such as aeroelastic instability.

These characteristics can be derived experimentally by wind tunnel testing or numerically via numerical simulation of the underlying basic equations of fluid dynamics depending on the accuracy required.

Complex flows around the airfoils are modelled using computational fluid dynamics (CFD) on a daily basis in the area of aerodynamics, since CFD tools have already reached an appropriate degree of maturity. However, this implies a large computing cost which may be infeasible in some cases presently. To address this constraint, the CFD solver might be replaced by a surrogate model that generates a quick forecast of the aerodynamic characteristics based on past simulations or wind tunnel data.

Machine learning techniques which are widely utilised in the field of artificial intelligence (AI) can provide significant aid in reducing the computing cost necessary for aerodynamic analysis. Machine learning techniques are currently being utilised in a variety of commercial sectors to exploit data in order to spot trends, find specific traits, patterns, and even forecast the future. One of the most important advantages of the machine learning method is its high working efficiency. As a result, many engineering problems particularly those involving numerical calculations can be solved in a relatively short period of time when certain and proper machine learning methods are implemented. However, the use of these concepts in the field of aerodynamics is still in its early phases.

1.1 Methodology

Methodology adopted for this work is application of Python language to create airfoil images using airfoil data repository, Numerical analysis to obtain aerodynamic characteristics of airfoils. Generating master dataset involving flow conditions to develop Convolutional Neural Network (CNN) model for predicting aerodynamic characteristics of airfoils.

2. NUMERICAL SIMULATION

The key to processing airfoil data is to prepare the input images. The CNN prediction model typically receives an image as input, a two-dimensional matrix for grayscale images and three two-dimensional matrices for colour images. The airfoil picture is grayscale image which can be used directly as input to the CNN model.

The flow conditions are three numerical parameters (Angle of Attack, Mach number, Reynold's number) that could not be utilised directly as input to the CNN prediction model. As a result, they are transformed into images in order to be identified.

2.1 Image Generation

As raw data, the UIUC Airfoil Data Site [1] which provides coordinates for almost 1600 mainstream airfoils ranging from the NACA 4-digit series to the Selig series is used. The raw coordinate pairs are converted into greyscale images.

The coordinate matrix is initially plotted as a contour image of 128 pixels by 128 pixels for each airfoil sample. Because aerodynamic performance of airfoils is very sensitive to subtle changes in shape, image size is set to (128x128) pixels instead of the normally smaller size, such as (28x28) in MNIST dataset or (32x32) in NIST36 dataset as it is critical to maintain a decent resolution to maintain prediction accuracy. However, because there is not too many subtle feature information in the airfoil, making the contour size with resolution as high as (256x256) in the dataset would be computationally costly [2]. When accuracy and computation are considered, an image size of (128x128) will be the optimum match for the application in airfoil data.

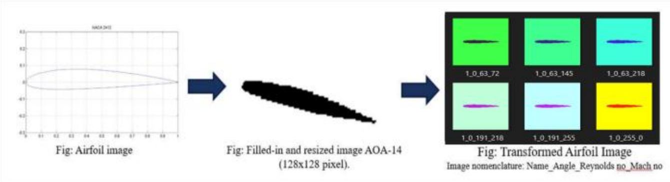

Fig -1: A snippet of Transformed Airfoil Images (TAIs) of an airfoil at various angles of attack, Reynolds number and Mach number.

The grayscale airfoil images are generated with the help of Python at various angles of attack. For this work, angle of attack varies from -5° to +20°. Later, these images are used to generate three channel images involving all the flow conditions. These generated images are Transformed Airfoil Image (TAI) which are complex representation of airfoil shape, angle of attack, Mach number and Reynolds number.

2.2 Ground Truth Calculation

The aerodynamic data is collected by a series of computer simulations. Given the expected variation in flow conditions and airfoil shapes, *JavaFoil* is selected as the simulation tool for determining aerodynamic characteristics.

JavaFoil is a simple software that analyses airfoils using numerous standard approaches [3].

For this work, range of Mach number considered is 0-0.7 with step size of 0.2 and for Reynolds number from 30000-1630000 with step size of 400000. A total of 102 airfoils comprising NACA 4 and NACA 6 series are used for model development. Output files in text format are obtained using *JavaFoil* for all the airfoils at different Reynolds and Mach number.

Mach = 0.4; Re = 30000; T.U. = 1.0; T.L. = 1.0
Surface Finish = 0; Stall model = 0; Transition model = 1; Aspect Ratio = 0; ground effect = 0

?	Cl	Cd	Cm 0.25	T.U.	T.L.	S.U.	S.L.	L/D	A.C.	C.P.
[°]	[-]	[-]	[-]	[-]	[-]	[-]	[-]	[-]	[-]	[-]
-5.0	-0.492	0.05047	0.003	0.800	0.007	1.000	0.009	-9.752	0.254	0.255
-4.0	-0.404	0.04399	0.002	0.732	0.010	1.000	0.013	-9.174	0.237	0.255
-3.0	-0.382	0.02229	0.004	0.697	0.029	1.000	0.995	-17.120	0.247	0.260
-2.0	-0.252	0.02134	0.003	0.662	0.504	1.000	0.908	-11.785	0.260	0.260
-1.0	-0.127	0.02094	0.001	0.626	0.548	1.000	0.929	-6.042	0.260	0.260
-0.0	-0.000	0.01760	-0.000	0.588	0.588	0.949	0.949	-0.000	0.260	0.250
1.0	0.127	0.02094	-0.001	0.548	0.626	0.929	0.993	6.042	0.260	0.260
2.0	0.252	0.02134	-0.003	0.504	0.662	0.908	0.995	11.785	0.260	0.260
3.0	0.382	0.02229	-0.004	0.029	0.697	1.000	0.995	17.138	0.247	0.260
4.0	0.404	0.04399	-0.002	0.010	0.732	0.013	0.995	9.174	0.237	0.255
5.0	0.492	0.05047	-0.003	0.007	0.800	0.009	0.995	9.752	0.254	0.255
6.0	0.566	0.06153	-0.003	0.005	0.892	0.007	0.995	9.195	0.256	0.255
7.0	0.612	0.07854	-0.003	0.005	0.907	0.006	0.995	7.793	0.271	0.255
8.0	0.606	0.10770	-0.004	0.005	0.920	0.006	0.995	5.626	0.243	0.256

Fig -2: Output File from *JavaFoil* for an airfoil for Re- 30000 and M- 0.4

2.3 Dataset Generation

The first and most important stage in determining the relation between aerodynamic characteristics and airfoil geometry using a neural network is to provide input data with suitable mathematical representation for training the neural network. Flow conditions, aerodynamic characteristics and airfoil images constitute the airfoil data.

Aerodynamic characteristics obtained from *JavaFoil* are in text format. All the text files of various airfoils at different Mach numbers and Reynolds number are read using Python to generate a consolidated file which acts as master dataset for development of the model. Fig 3 shows a snippet of the master dataset.

Airfoil No.	Reynolds Number	Mach Number	Angle	C_L	C_L/C_D	C_D
24	430000	0.2	-5	-0.085	-10.536	0.00807
24	430000	0.2	-4	0.036	4.593	0.00778
24	430000	0.2	-3	0.158	20.827	0.00759
24	430000	0.2	-2	0.281	37.59	0.00747
24	430000	0.2	-1	0.404	50.964	0.00792
24	430000	0.2	0	0.526	80.781	0.00651
24	430000	0.2	1	0.647	106.851	0.00605
24	430000	0.2	2	0.768	121.27	0.00633
24	430000	0.2	3	0.888	132.492	0.0067
24	430000	0.2	4	1.006	100.595	0.01
24	430000	0.2	5	1.121	104.02	0.01078
24	430000	0.2	6	1.23	102.761	0.01197
24	430000	0.2	7	1.334	97.693	0.01365
24	430000	0.2	8	1.417	70.806	0.02001
24	430000	0.2	9	1.488	65.282	0.02279
24	430000	0.2	10	1.54	59.908	0.0257
24	430000	0.2	11	1.568	53.127	0.02951
24	430000	0.2	12	1.569	45.325	0.03462
24	430000	0.2	13	1.539	36.456	0.04221
24	430000	0.2	14	1.465	25.806	0.05677

Dataset size – 66300

Training samples- 47736

Validation samples- 11934

Test samples- 6630

Re range- 30000-1630000

Mach range- 0-0.7

Fig -3: Master Dataset (csv format)

3. MODEL ARCHITECTURE

An initial model architecture was selected and later OPTUNA [4], a hyperparameter optimization framework was used to automate hyperparameter selection. The hyperparameters optimized included dropout rate, batch size, learning rate and fully connected layer input dimension.

Table -1: Hyperparameters

Hyper-parameter	
Learning Rate	0.00506
Batch Size	50
Drop-out Rate	0.2
FC-2 in-features	650

Final model architecture is as follows:

- 5 Convolution layers

  I. Layer 1

  2D convolution with input channel = 3, output channel = 20 and kernel size of 5x5. Followed by batch normalization and max pooling with kernel size of 2x2. Lastly, ReLU as activation function.

  II. Layer 2

  2D convolution with input channel = 20, output channel = 40 and kernel size of 3x3. Followed by batch normalization and max pooling with kernel size of 2x2. Lastly, ReLU as activation function.

  III. Layer 3

  2D convolution with input channel = 40, output channel = 80 and kernel size of 3x3. Followed by batch normalization and max pooling with kernel size of 2x2. Lastly, ReLU as activation function.

  IV. Layer 4

  2D convolution with input channel = 80, output channel = 160 and kernel size of 3x3. Followed by batch normalization and max polling with kernel size of 2x2. Lastly, ReLU as activation function.

  V. Layer 5

  2D convolution with input channel = 160, output channel = 200 and kernel size of 2x2. Followed by batch normalization, max polling with kernel size of 3x1 and drop out set to 0.2. Lastly, ReLU as activation function.

- 2 Fully connected layers

  I. First fully connected (FC) layer has linear function which transforms the incoming data. Input features = 1800 and output features = 650. Drop out is set to 0.2 and ReLU as activation function.

  II. Second FC layer has input features = 650 and output features = 1.

4. MODEL DEVELOPMENT

4.1 Model for C_L Prediction

Model is trained on the training data which constitutes 72% of the master dataset and its performance is validated after each epoch using validation dataset which constitutes 18% of the master dataset. After the training is completed, model is evaluated on test dataset which is 10% of the master dataset. During the training process the model converged after 100 epochs, as losses and accuracies of each subsequent epoch remained same. Also, further training could overfit the model causing it to lose its generalization capabilities.

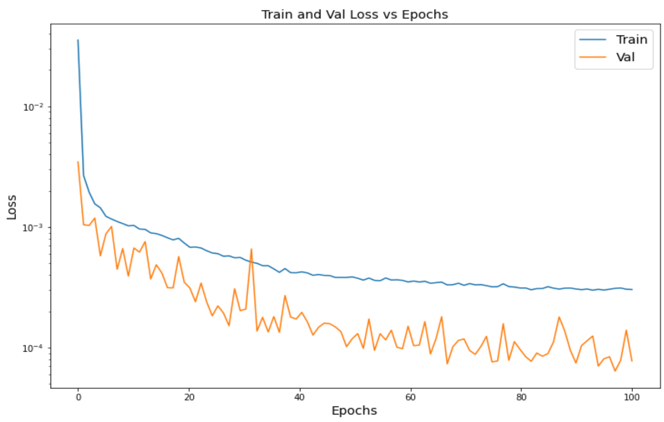

Fig -4: Loss Plot

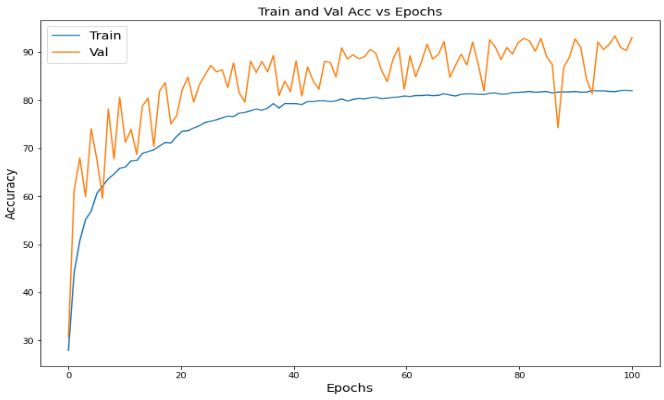

Fig -5: Accuracy Plot

Test accuracy obtained is 92.32%. The confusion matrix shows the prediction capability of the model.

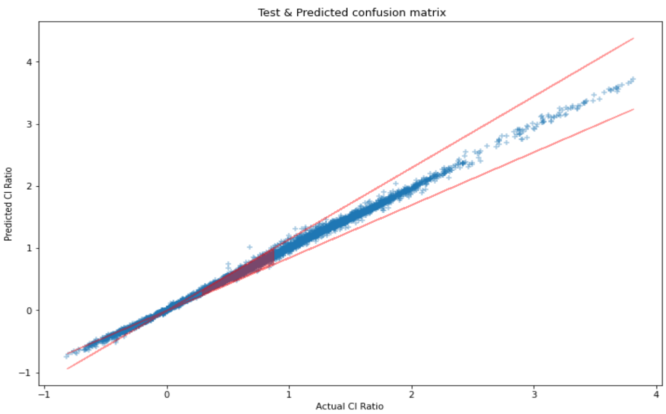

Fig -6: Confusion Matrix

4.2 Model for C_D Prediction

During the training process the model converged after 150 epochs, as losses and accuracies of each subsequent epoch remained same. Also, further training could overfit the model causing it to lose its generalization capabilities.

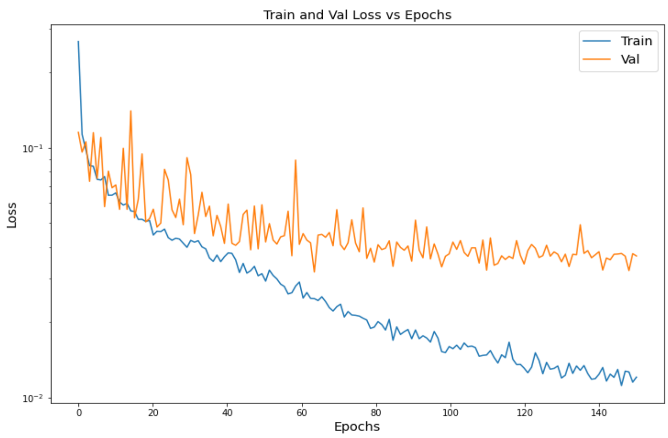

Fig -7: Loss Plot

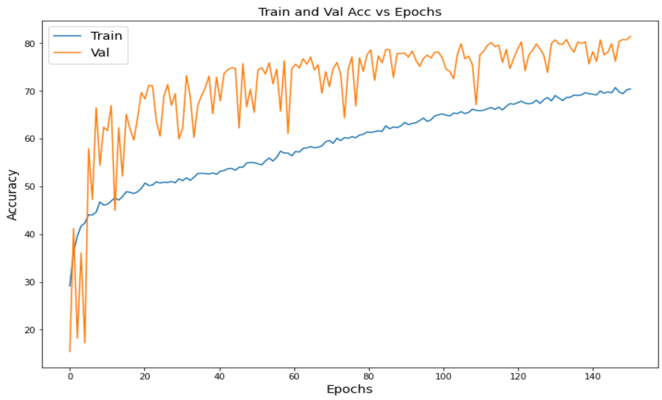

Fig -8: Accuracy Plot

Test accuracy obtained is 81.42%. The confusion matrix shows the prediction capability of the model.

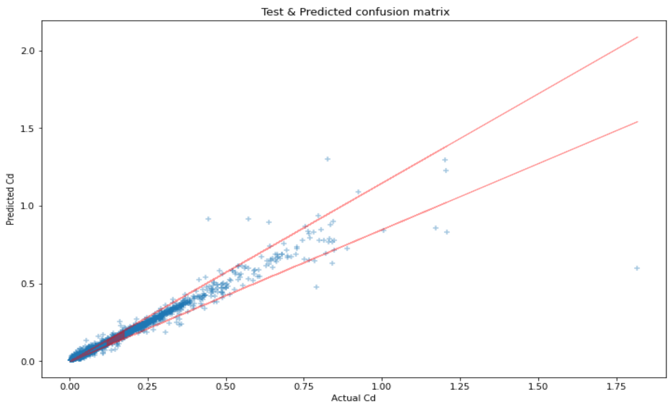

Fig -9: Confusion Matrix

4.3 Model for $\frac{C_L}{C_D}$ Prediction

During the training process the model converged after 180 epochs, as losses and accuracies of each subsequent epoch remained same. Also, further training could overfit the model causing it to lose its generalization capabilities.

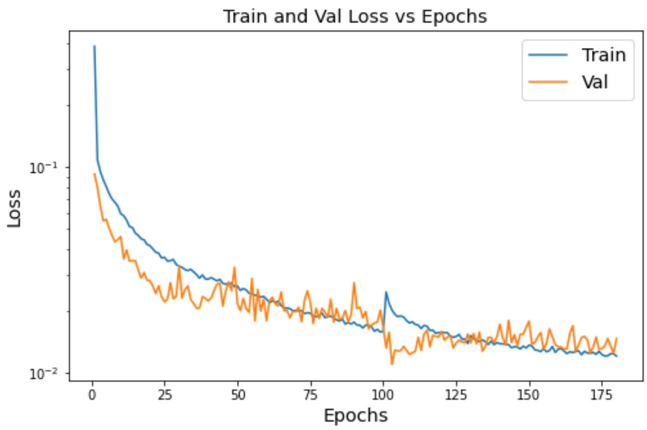

Fig -10: Loss Plot

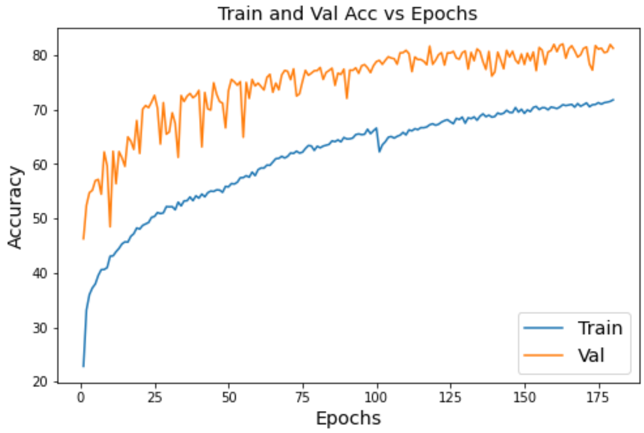

Fig -11: Accuracy Plot

Test accuracy obtained is 81.29%. The confusion matrix shows the prediction capability of the model.

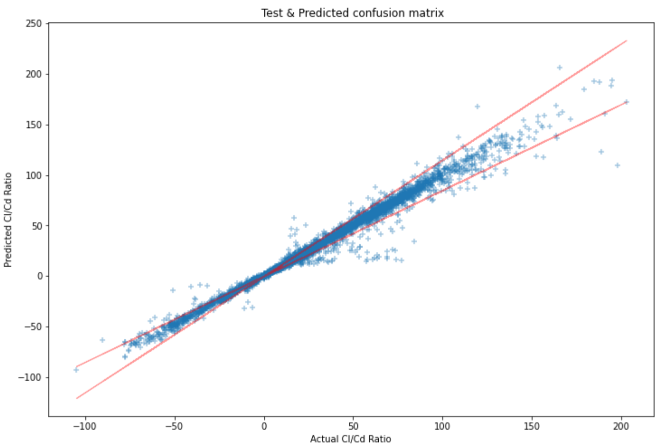

Fig -12: Confusion Matrix

These models developed can be used to predict each aerodynamic characteristic individually or all the models can be used at the same time to make inference. The models allow to choose any Mach number from 0-0.7, Reynold's number from 30000- 1630000 and work on any airfoil.

5. MODEL INFERENCE

5.1 Prediction of Aerodynamic Characteristics of Airfoils

After the model development is completed, the next step is to put the model in inference mode to start using it for predicting the aerodynamic characteristics of airfoils. Input file of any airfoil is created using python for specified flow conditions which allows the model to call the input images and later it predicts the output and stores it in the same file.

Two cases are considered for analyzing model performance:

1. Airfoil which is part of the dataset (NACA 6409)
2. Airfoil which is not part of the dataset (NACA 13013)

➢ **NACA 6409**

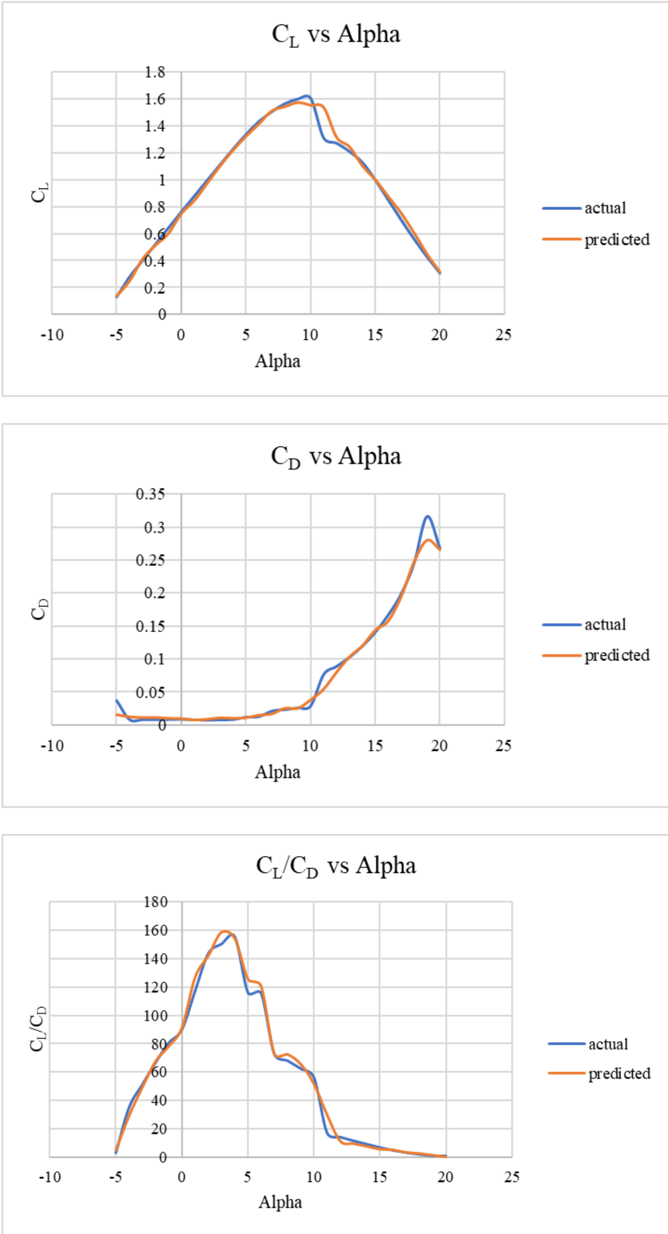

Fig -13: Inference Graphs for NACA 6409 airfoil

Flow conditions- Reynold's No- 430000, Mach No- 0.2

The graphs of inference obtained from the models and the actual data shows that the model has predicted aerodynamic characteristics quite accurately.

- **NACA 13013**

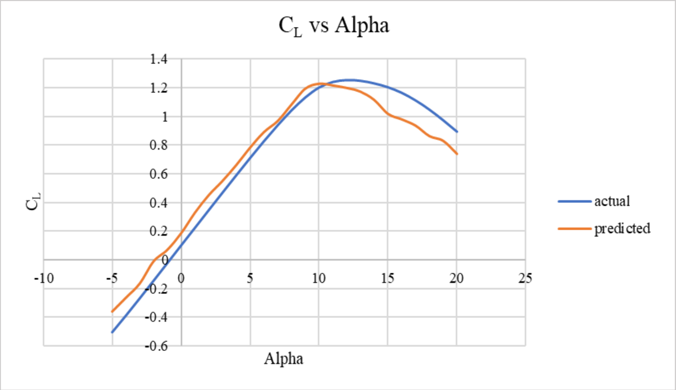

Fig -14: Inference Graphs for NACA 13013 airfoil

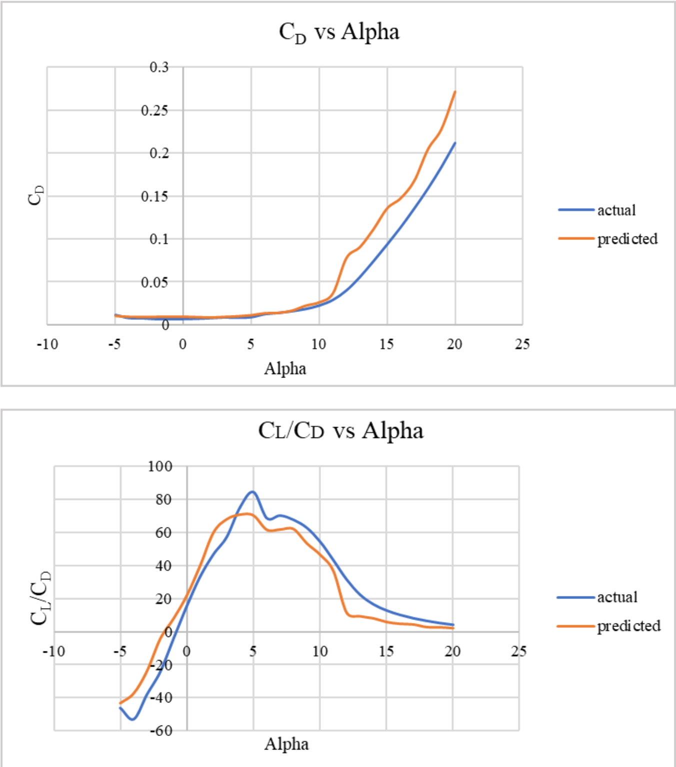

Fig -15: Inference Graphs for NACA 13013 airfoil

Flow conditions- Reynold's No- 430000, Mach No-0.2

It can be observed that model has good accuracy on airfoils which were not part of the dataset. It took 0.66 second for the model to make predictions (All aerodynamic characteristics). However, there is large error is predicting $\frac{C_L}{C_D}$ compared to other parameters.

A batch analysis involving prediction of aerodynamic characteristics of airfoils using the model and XFLR5 [5] shows, CNN model is 50 times efficient compared to XFLR5.

Table -2: Efficiency Comparison

Batch Size (sec)	CNN (sec)	XFLR5 (sec)
10	4.48	148.31
20	8.56	270.57
30	12.98	631.26
50	20.94	1256.4
100	40.45	2507.13

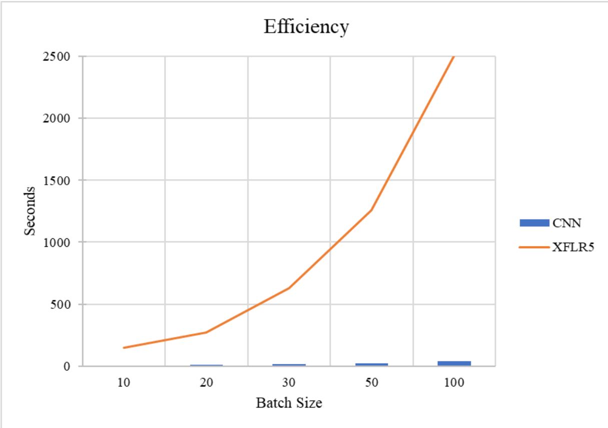

Fig -16: Efficiency Plot

6. CONCLUSION

This work contributes toward development of a surrogate model which can predict aerodynamic characteristics of airfoil subjected to various flow conditions. The developed CNN model can predict C_L, C_D and $\frac{C_L}{C_D}$ for different airfoils with the help of generated Transformed Airfoil Images (TAI) using flow conditions and coordinates of the airfoil.

The CNN model has high level of accuracy for airfoils which were part of dataset as well as airfoils which are new to the model. For C_L model, accuracy is 92.32%, C_D model it is 81.42% and for $\frac{C_L}{C_D}$ it is 81.29%. It took 100 epochs to train C_L model, 150 epochs to train C_D model and 180 epochs to train $\frac{C_L}{C_D}$ model.

Model inference shows the capability of the model in prediction of the aerodynamic characteristics. It took 0.66 second for the model to make predictions (All aerodynamic characteristics) which is highly efficient compared to CFD analysis as well as batch mode of XFLR5. CNN model is 50 times efficient compared to batch mode of XFLR5. The models work on any airfoil. They allow to choose any Mach number from 0-0.7 and Reynold's number from 30000-1630000.

However, there is high error is predicting $\frac{C_L}{C_D}$ compared to other parameters for airfoils due to high dispersion of the data.

The performance of the models can be improved by increasing the size of the dataset along with the application of Virtual Machines to train the model, due to high training time requirement of the model.

However, in its current state the developed models are ready to predict the aerodynamic characteristics with 15% relative tolerance and it can help to curb the time taken during preliminary stage of airfoil selection.

REFERENCES

[1] UIUC Airfoil Data Site, Department of Aeronautical and Astronautical Engineering University of Illinois at Urbana-Champaign, 1996.

[2] Haolin Liu, Zi Li, and Felix Lu. An Airfoil Aerodynamic Parameters Calculation Method Based on Convolutional Neural Network. 2019.

[3] www.mh-aerotools.de/

[4] www.optuna.org/

[5] www.xflr5.tech/xflr5.htm

[6] Anderson Jr., John D. 2010. Fundamentals of Aerodynamics, 5th Edition, McGraw- Hill.

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

[8] Russell Reed and Robert J MarksII. Neural smithing: supervised learning in feed forward artificial neural networks. Mit Press, 1999.