

Estimating Pitching Airfoil Kinematics From Downstream Flowfield Measurements using Convolutional Neural Networks

Kunwoo Lee * and Scott T. M. Dawson †
Illinois Institute of Technology, Chicago, IL, 60616

Convolutional neural networks (CNNs) have recently demonstrated significant power in analyzing fluid flows. Our goal is to introduce a novel CNN approach for estimating airfoil pitching kinematics based on a small, instantaneous snapshot from the downstream region of the unsteady wake. Initially, we concentrated on a sinusoidal pitching airfoil created from direct numerical simulation (DNS). The model exhibited exceptional accuracy in predicting the airfoil's base angle of attack and pitching frequency. Encouraged by this success, we subjected the model to additional challenges by reducing the size of the input image in two ways: 1) decreasing the measuring field area and 2) downsampling the image. In the former approach, we reduced it until the field edge length was 0.8 times the chord length. Despite this substantial reduction, the accuracy remained at 90.631%. In the latter approach, downsampling preserved accuracy even better. The accuracy remained at 98.458% even in the most extreme case. Our investigation revealed that the measuring field area dominated the prediction accuracy. To further challenge model's capabilities, we tested it on chirp signal pitching data, also created through DNS. Initial attempts to predict chirp signal data using models trained on sinusoidal pitching data were unsuccessful. The model struggled to identify commonalities between flows with the same base angle of attack and frequency when the frequency varied. However, when the model was trained with varying frequency data, it achieved high accuracy (95.5%) in predicting the kinematics, maintaining strong performance on sinusoidal cases as well (99.8%). Even after training with chirp signal pitching data, the model demonstrated a lack of robustness when faced with chirp signal pitching data with a different rate of frequency increase. This project exhibited high potential of CNNs, showing highly accuracy, but raised need for further investigation on improving robustness of the model.

Nomenclature

α_0 = the base angle of attack in oscillation ($^{\circ}$)

*Undergraduate Research Assistant, MMAE department

†Assistant Professor, MMAE Department, AIAA Senior Member.

| | | |
|--------------|---|---|
| α | = | the instantaneous angle of attack ($^{\circ}$) |
| f_p | = | dimensionless frequency (constant) |
| ω | = | instantaneous dimensionless frequency (varying frequency) |
| c | = | chord length (meter) |
| U_{∞} | = | freestream velocity (meter/sec) |
| α_p | = | pitching amplitude ($^{\circ}$) |
| u_x | = | Streamwise velocity (x -velocity) |
| u_y | = | Wall-normal velocity (y -velocity) |

I. Introduction

RAPID motion of airfoils can cause substantial wake fluctuations, making wake behind the airfoil unsteady. It is important to understand the unsteady aerodynamics to control any system within or influenced by the wake. This is challenging as there are numerous parameters that govern the system which changes along time, interconnected to each other in a complicated manner.

The unsteady aerodynamics analysis can be done by solving the Navier-Stokes equation. However, the complexity of the equation makes it beyond current computational capabilities in most practical cases. To resolve this issue, one branch of study focused on dimensionality reduction and reduced order modeling. Traditional approaches in this branch include Proper Orthogonal Decomposition (POD)[1][2], Dynamic Mode Decomposition (DMD)[3], and Koopman analysis[4]. The problem with these approaches is that they mainly deal with linear domains, which inevitably cause information loss. This can be a severe issue for unsteady flows, which often require accurate prediction of transient phenomena.

Based on the universal approximation theorem, neural network (NN) models can approximate any given function. Due to its structural nature, neural network models are not limited to linear domains. This led to the idea that a sufficiently deep neural network can be optimized to a certain function that can outperform the traditional approaches. Ref. [5] performed dimensionality reduction and DMD using neural network. Neural network autoencoder was able to perform dimensionality reduction and reconstruction with average 93% reconstruction accuracy. Furthermore, the autoencoder-LSTM (long short-term memory) framework was proven to outperform the DMD method, and the performance gap was much higher for complex cases with multi-frequency and extreme events phenomena. Ref. [6] focused on reduced order modeling and reconstruction of advection-dominated systems, which are challenging for the traditional method of POD & Galerkin projection due to their high dimensionality. Convolutional autoencoder (CAE) and LSTM was used, showing that the proposed method was novel both in stability and convergence time. Ref. [7] proposed a new approach for the prediction of drag for various shapes of airfoils using convolutional neural network (CNN).

Another strength of the NN-based approach is in classification. Neural network model is superior in finding patterns that are too complex to model, due to its model-free nature. In this context, wake classification has been in spotlight as well, hoping to predict the upstream disturbance just by classifying the wake. Ref. [8] examined to give insight on wake similarity between various airfoil kinematics by NN-based clustering. Two dimensional convolutional layers were used to extract critical features of wake, followed by LSTM network classifying the input sequence, achieving an average of 90% accuracy.

There have been largely two types of neural network models that gained attention: convolutional neural networks (CNNs) and recurrent neural networks (RNNs). CNNs were proven to be powerful in extracting features from spatial data, while RNNs have its strength in sequential data. Fluid-related quantities are often spatial and sequential data at the same time. Many approaches leveraged CNNs and RNNs together in this manner, such as Ref. [6]. Using CNNs only in fluid problems may be beneficial due to faster prediction time, as CNNs are typically faster than RNNs. CNNs allow parallel processing due to its structural nature, while RNNs process its input in sequential manner, one element at a time. This may be critical when applying RNNs for system that requires fast feedback for control.

The weakness of CNNs is that it requires vast, high-quality image data. This is problematic especially for fluid application, as it is relatively difficult to create vast, high-quality data of fluid. Nowadays, advanced techniques to create such images are keep evolving. Starting from the development of Particle Image Velocimetry (PIV), many researchers in the field are striving for super-resolution imaging [9] and denoising algorithms [10], giving the possibility to overcome the weakness of CNNs.

To summarize, there is a promising possibility for CNNs to perform well in fluid problems for complex, high-dimensional flows with inherent nonlinearity. Furthermore, CNNs are suitable for applications that require fast feedback for control. In this background, we focused on an inverse problem to estimate the airfoil pitching kinematics using CNN from the small, instantaneous velocity field measured in the wake far from the airfoil. The second section of the paper is dedicated to give basic introduction to CNNs, along with the third section describing the database used to feed our model. In the fourth section, the basic setup for the estimation testing is addressed. Then, listed the cases explored are listed, with detailed explanation. The cases are largely classified into the performance of CNNs on sinusoidally pitching airfoil, and chirp signal pitching prediction. Finally, the result and conclusion are given in section 6 & 7. The code used for this paper is available at <https://github.com/KunwooLeeKay/Estimating-Airfoil-Pitching-Kinematics>.

II. Basic Component of Convolutional Neural Network

CNNs models are trained by optimizing *filters*. The basic framework of CNNs is shown in Fig. 1. The filters set the area of input image to be read in each *convolutional operation*, where it outputs summation of element-wise multiplication between the area read and filter. Convolutional operation is done for whole image as the filters move inside the image, in step size of *stride*. The filtered image after convolutional operation then goes through activation

function, forming *feature map*.

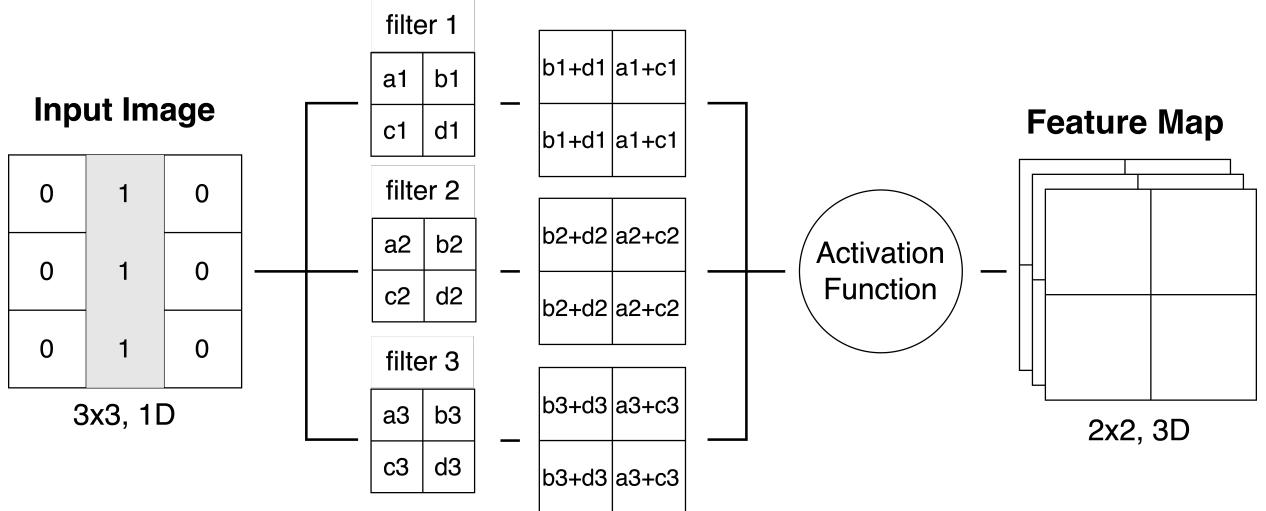


Fig. 1 The basic framework of the convolutional layer that with 3x3 input image and three 2x2 filters.

III. Database Description: Direct Numerical Simulation

The database described in Ref. [11] was used. It is available at <https://deepblue.lib.umich.edu/data/collections/kk91fk98z?locale=en>. Example snapshot of this database is shown in Fig. 2. The corresponding database is created by numerically solving the Navier-Stokes equation using immersed boundary projection method, or direct numerical simulation (DNS). It models sinusoidal pitching airfoil in low Reynolds number flow ($Re = 100$). Reynolds number was defined from freestream velocity (U_∞) and chord length (c). The motion of the airfoil follows Eq. (1).

$$\alpha(t) = \alpha_0 - \alpha_p \sin(2\pi f_p t) \quad (1)$$

The database provides two categories of α_0 , 25° and 30° . The pitching amplitude α_p was set constant to 5° . The database provides 8 categories of f_p in set $\{0.05, 0.1, 0.2, 0.25, 0.3, 0.35, 0.4, 0.5\}$. Note that f_p is dimensionless frequency, obtained by $f_p = fc/U_\infty$. The velocity measurement was taken every 10 dimensionless timesteps, defined as $0.01c/U_\infty$. The stream-wise domain length was $12 \times c$, and wall-normal length of domain was $6 \times c$, making 600×300 grid points. The outputs used to train CNN were u_x (streamwise velocity) and u_y (wall-normal velocity). A more detailed explanation of the dataset can be found in [11].

Simulation for chirp signal pitching was done as well for further investigation on performance of CNN. Aforementioned method of DNS was applied, but varying frequency ω substituted constant frequency f_p . The airfoil pitching motion is described in Eqs. (2) (3). Note that $f_1 = 0.5$ and $f_0 = 0.05$ in Eq. (2), making the dimensionless frequency to increase

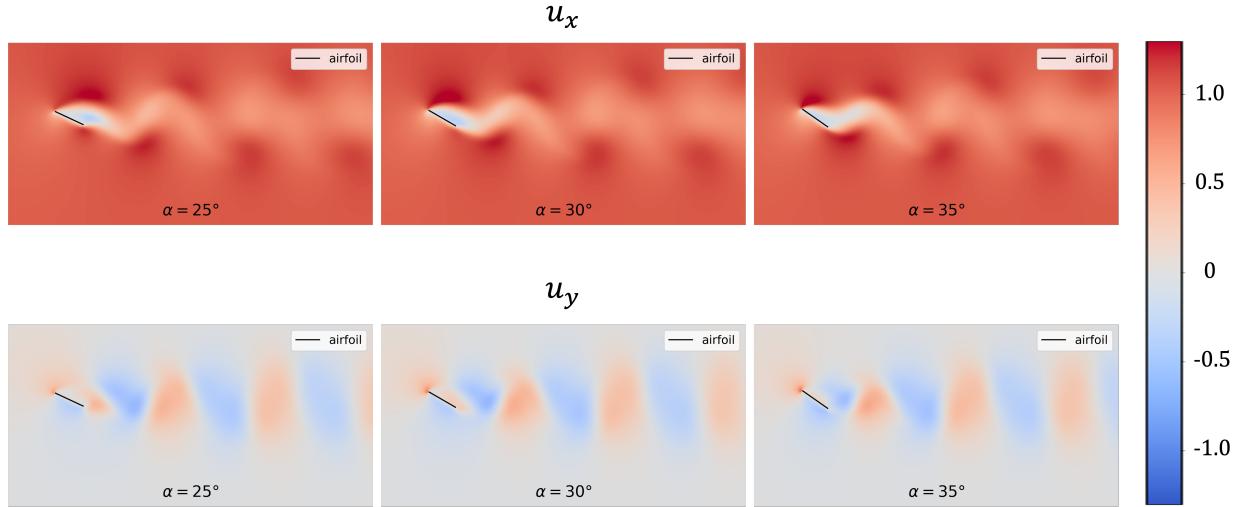


Fig. 2 The example of instantaneous dimensionless velocity field for U_x and U_y at 25° , 30° , and 35° . Base angle of attack = 30° , pitching amplitude = 5° , dimensionless pitching frequency = 0.25

linearly from f_0 to f_1 in time span of t_{end} . The value of t_{end} was tested for two cases, 6000 and 9000.

$$\omega(t) = f_0 + \frac{t}{t_{end}}(f_1 - f_0) \quad (2)$$

$$\alpha(t) = \alpha_0 + \alpha_p \cos(2\pi\omega t) \quad (3)$$

IV. Basic Setup for Airfoil Pitching Kinematics Prediction

A. Making Input Image from Database

The schematic process to create input image for CNN is shown in Fig. 3. As shown in Fig. 3, a certain size of window for each u_x and u_y was taken from the wake far from the airfoil. Then they were concatenated vertically to form a square image. The size of input image is denoted by ratio throughout this literature. In the case of Fig. 3, the original image size is 600×300 for both u_x and u_y . The size of window is 200×100 and concatenated image is 200×200 . Comparing the original image and the input image, the length of sides of the rectangular window are shrunk in ratio of 1/3. This is named 1/3 ratio input image size.

B. CNN Architecture

CNN architecture shown in Fig. 4 was used throughout the testing. There were seven convolutional layers (conv2d layer) followed by a fully connected layer and the output layer.

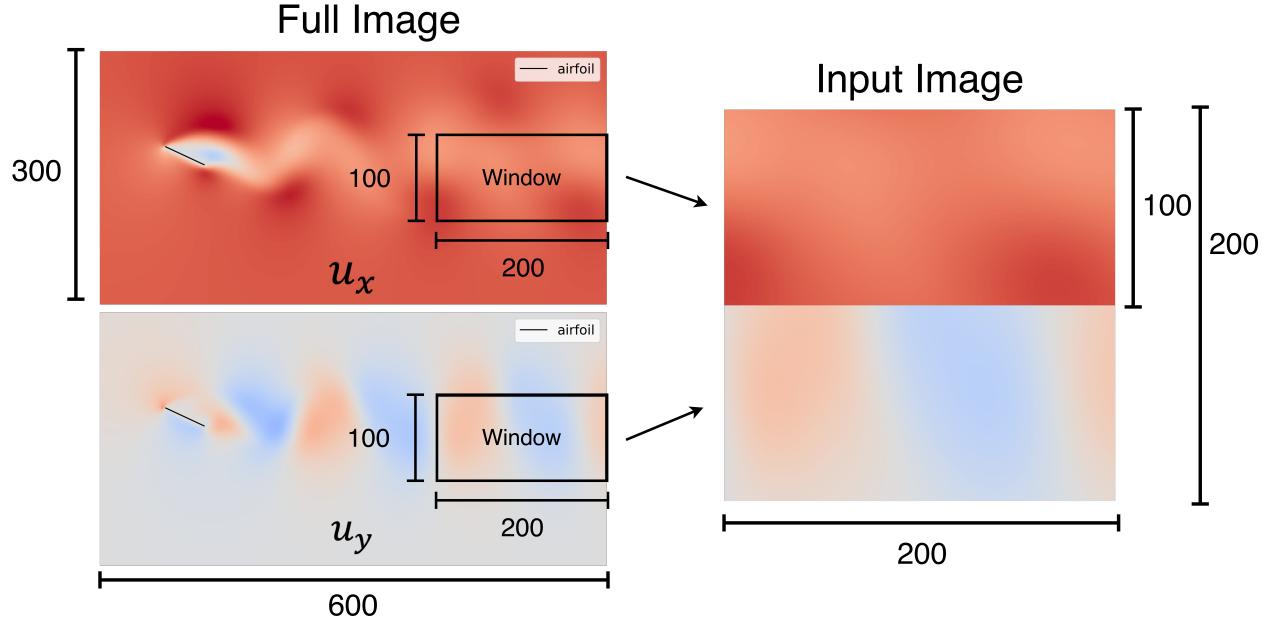


Fig. 3 The schematic illustration of the process to create a square-shaped input image.

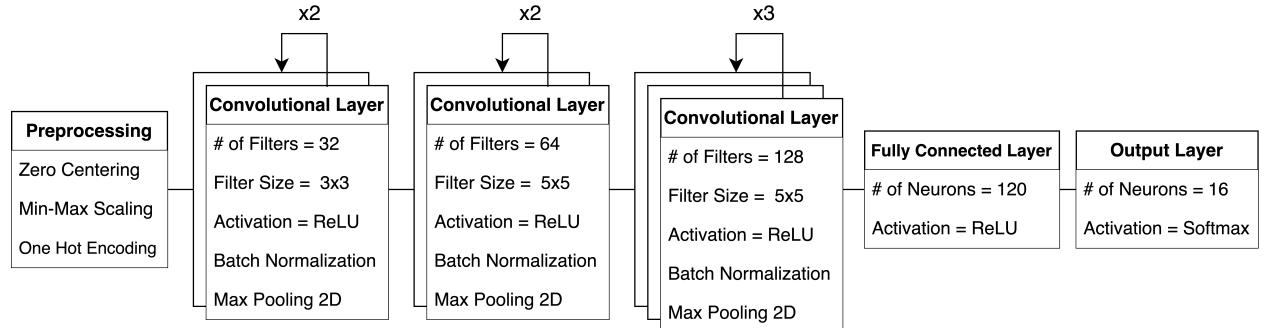


Fig. 4 The schematic diagram of CNN architecture used for the prediction.

C. Performance Evaluation

The performance of the CNN model was evaluated mainly on accuracy. The confusion matrices were examined as well to check any problems in the model or any possible insights. Due to the random nature of the training neural networks, repeating the training produces a slightly different model. Testing, the range of accuracy was not subtle enough to ignore the effect of randomness, so all the evaluation of performance was done on 10 CNNs trained in different random seeds, and we used the averaged results for every case.

V. Cases Explored

In this literature, we aim to explore the usage of CNN to estimate the pitching kinematics of an airfoil. The following cases are to be discussed:

A. Sinusoidal Pitching Airfoil Kinematics Prediction

- 1) Challenging CNN Model - Effect of Reducing Input Image Size
 - 1) Reduction 1. Reducing Measuring Field Area (Window Size)
 - 2) Reduction 2. Downsampling
- 2) Benchmark of Estimation for CNN
 - 1) Performance Comparison between CNN and Sparse Basis Selection

B. Chirp Signal Pitching Airfoil Kinematics Prediction

- 1) Test 1. Chirp Signal Prediction with Model Trained on Sinusoidal Pitching Data
- 2) Test 2. Chirp Signal Prediction with Model Trained on Sinusoidal + Chirp Signal Pitching Data
- 3) Test 3. Out-of-Distribution Chirp Signal Prediction with Model Trained on Sinusoidal + Chirp Signal Pitching Data

A. Sinusoidal Pitching Airfoil Kinematics Prediction

1. Challenging CNN Model - Effect of Reducing Input Image Size

Two methods of image size reduction technique was used to test the effect of input image size, denoted reduction 1 and reduction 2. First, reduction 1 was reducing measuring field area (or window size), meaning that we kept decreasing the size of the black rectangle shown in Fig. 3. As mentioned before, reduction is quantified in ratio, which corresponds to in what ratio the length of the sides were reduced. The tested ratios are $1/3, 1/4, 1/5, 1/6, 1/7, 1/8, 1/9, 1/10, 1/11, 1/12, 1/13, 1/14$, and $1/15$, which corresponds to input images of $200 \times 200, 150 \times 150, 120 \times 120, 100 \times 100, 86 \times 86, 76 \times 76, 66 \times 66, 60 \times 60, 54 \times 54, 50 \times 50, 46 \times 46, 42 \times 42$, and 40×40 , respectively.

Reduction 2 was simplified image downsampling. The schematic process is shown in Fig. 5. (Note that this illustration shows downsampling process for magnified fragment. The full input image is made by performing downsampling for whole original image.) The downsampling was done by skipping a number of pixels before collecting the next one from the original image. The number of pixels skipped is named *strides*. The images from reduction 1 were used as the original image before downsampling. Here, we denote the size of the image before downsampling as the base window ratio. For example, if the base window ratio is $1/3$ and stride = 1, then the input image size becomes that equivalent to window ratio of $1/6$, making input image 100×100 . If the base window ratio is $1/4$ and stride = 1, it becomes that equivalent to window ratio of $1/8$, making input image 76×76 . The base window ratios of $1/3, 1/4, 1/5, 1/6$, and $1/7$ were used, and all the tested cases are listed in table 1 for better understanding.

All data used to train CNN were split into train, validation, and test data with ratio $70 : 15 : 15$. The number of images for train, validation, and test were 7856, 1680, and 1680, respectively. The splitting was done randomly but was stratified with categories. Then, three types of preprocessing were done: zero-centering, scaling, and weighting.

- Zero-centering: Mean image was calculated from training data. Then, the mean image was subtracted from training, validation, and test data.

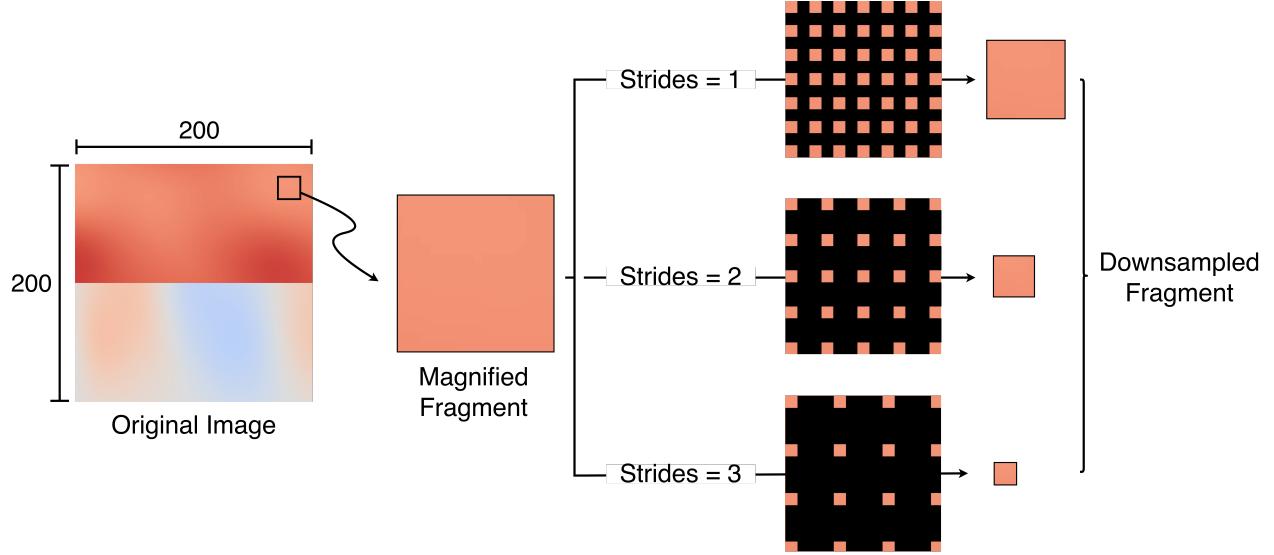


Fig. 5 The schematic of the process to create downsampled images according to strides value. The example is shown for a fragment of the whole image for better understanding.

Table 1 Tested cases for test 2

| Base Window Ratio | Strides | Side Length |
|-------------------|---------|-------------|
| 1/3 | 1 | 100 |
| 1/3 | 2 | 66 |
| 1/3 | 3 | 50 |
| 1/3 | 4 | 40 |
| 1/4 | 1 | 76 |
| 1/4 | 2 | 50 |
| 1/5 | 1 | 60 |
| 1/5 | 2 | 40 |
| 1/6 | 1 | 54 |
| 1/7 | 1 | 42 |

- Scaling: The range of value for u_x was significantly larger than u_y . Therefore, u_x and u_y velocity were scaled using minimum and maximum values of them, as in Eq. (4). The minimum and maximum values were calculated from training data, and these values were used to scale all training, validating, and testing data.
- Weighting: There were more data with $\alpha_0 = 30^\circ$ than $\alpha_0 = 25^\circ$. The loss from 25° cases was weighted by the factor of 2.5 when training CNN.

$$u_x = u_x / (\max(u_x) - \min(u_x)) \quad (4)$$

$$u_y = u_y / (\max(u_y) - \min(u_y))$$

2. Benchmark of Estimation for CNN

As a benchmark for comparison, we also use an alternative classification methodology, which is described here.

We develop a model that can identify the parameters α_0 and f_p . We do this by applying a machine learning method based on the ideas that are described and applied to reconstruction and parameter estimation in cylinder flow in Refs. [12, 13], as also described in Ref. [14]. We start by performing proper orthogonal decomposition on data from each simulation and assemble a truncated set of r POD modes for each case into columns of a matrix $\Phi_r^{(j)}$, where the superscript (j) indexes each simulation. For this example, we only consider data from the pitching cases ($\alpha_P = 5^\circ$), and use 401 snapshots from each case. These matrices are concatenated into a global library of modes

$$\Phi_L = \begin{bmatrix} \Phi_r^{(1)} & \Phi_r^{(2)} & \dots & \Phi_r^{(M)} \end{bmatrix}, \quad (5)$$

where $M = 16$ is the total number of pitching airfoil cases considered. A given snapshot x from simulation (j) may be approximated by $\Phi_r^{(j)} a^{(j)}$, where a is a vector of POD coefficients. Equivalently, we can also use the full library to write

$$x \approx \Phi_r^{(j)} a^{(j)} = \Phi_L a_L, \quad (6)$$

where a_L contains nonzero entries in rows corresponding to the set of columns of Φ_L containing $\Phi_r^{(j)}$.

We consider measurements defined by $y = Cx$, where each of the rows of C specifies which value(s) of x are used for a given measurement. In this case, we will use point measurements of one velocity component, so each row of C will contain zeros and a single entry of 1. We may use compressed sensing principles [? ?] to reconstruct the state x from measurements y . This is achieved by solving the convex optimization problem

$$\hat{a}_L = \operatorname{argmin} \|a_L\|_1 \text{ such that } \|C\Phi_L a_L - y\|_2 < \epsilon, \quad (7)$$

from which the full state x can be obtained from Eq. (6). Minimizing the l_1 norm promotes a sparse solution for \hat{a}_L , where most of its entries are zero. Ideally, the nonzero entries of the reconstructed coefficients \hat{a}_L would correspond to the POD modes associated with the correct simulation parameters, allowing for the frequency and base angle of attack to be identified.

B. Chirp Signal Pitching Airfoil Kinematics Prediction

The input image was made the way described in Fig. 3. Unlike the sinusoidal case, only one case of input image size was tested. The tested case was window size (measuring field area) of 200×200 with no stride. The frequency for chirp signal is increasing linearly, as in Eq. (2). To test on classification model we have been working on, we categorized the frequency into 8 categories: $\{0.05, 0.1, 0.2, 0.25, 0.3, 0.35, 0.4, 0.5\}$. This was done by taking the average values in two

consecutive categories and using them as the boundary values. For example, frequency in range $0.05 \leq \omega < \frac{0.05+0.1}{2}$ was categorized into label 0.05, frequency in range $\frac{0.05+0.1}{2} \leq \omega < \frac{0.1+0.2}{2}$ was categorized into label 0.1, etc. The boundary of frequency categorization is illustrated in Fig. 6.

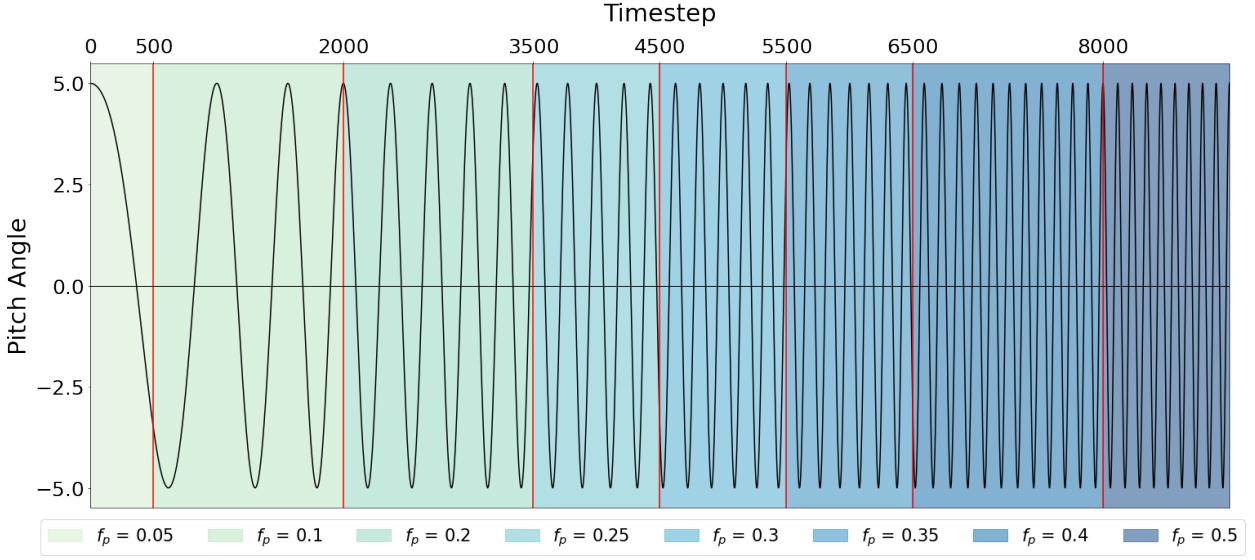


Fig. 6 The range of dimensionless frequency for the categorization of chirp signal.

Two types of preprocessing were done on chirp signal pitching data: zero-centering and scaling. They were done in the same manner as in sinusoidal pitching, defining the mean image and scaler from training data and applying it to the validation and test data. Note that for chirp signal pitching prediction, the ratio for train : validation : test is different for each case as there are cases where training data and test data is not from same type of simulation.

1. Test 1. Chirp Signal Prediction with Model Trained on Sinusoidal Pitching Data

The model trained from sinusoidal pitching data of reduction 1 was used (the ratio was 1/3). So, all the data generated for the chirp signal of $t_{end} = 9000$ was used for testing, making training : testing = 7856 : 1798.

2. Test 2. Chirp Signal Prediction with Model Trained on Sinusoidal + Chirp Signal Pitching Data

In test 2, a new model was trained using both sinusoidal data and chirp signal data. So the data split was done by train : validation : test = 70 : 15 : 15, making 9109 : 1953 : 1952. Note that in test 2, the ratio of sinusoidal : non-sinusoidal was about 6:1. This will be an imbalanced dataset, and this is intended as test 2 was done to make the model learn the sense of varying frequency while being able to classify sinusoidal pitching well.

3. Test 3. Out-of-Distribution Chirp Signal Prediction with Model Trained on Sinusoidal + Chirp Signal Pitching Data

In test 3, the dataset for training and testing was different from in test 1. The training data was created in test 2, which was 9109 data, and all the data generated for chirp signal of $t_{end} = 6000$ was used, which was 1198 data.

VI. Result

A. Sinusoidal Pitching Airfoil Kinematics Prediction

1. Reduction 1. Reducing Window Size

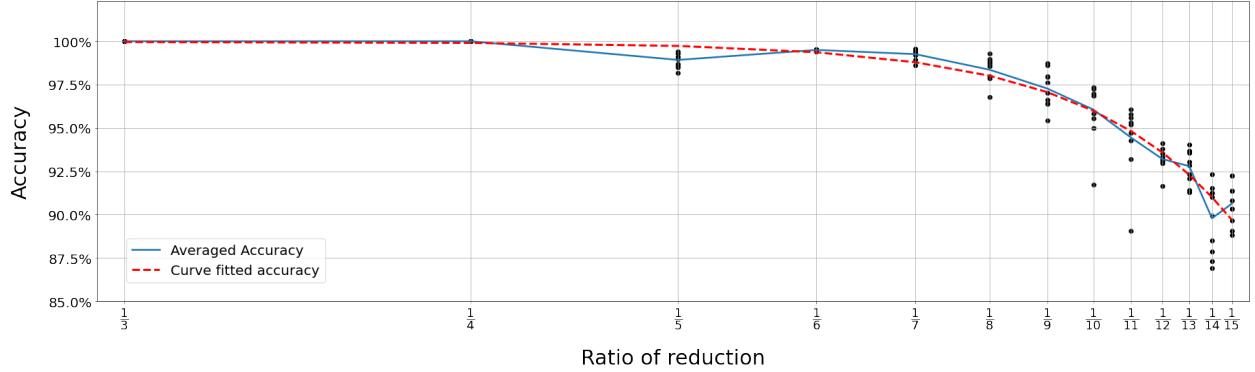


Fig. 7 The result for the reduction 1.

Figure.7 shows the result of reduction 1. The x axis represents the ratio of reduction, which are $1/3, 1/4, 1/5, 1/6, 1/7, 1/8, 1/9, 1/10, 1/11, 1/12, 1/13, 1/14$, and $1/15$. They correspond to square images of $200 \times 200, 150 \times 150, 120 \times 120, 100 \times 100, 86 \times 86, 76 \times 76, 66 \times 66, 60 \times 60, 54 \times 54, 50 \times 50, 46 \times 46, 42 \times 42$, and 40×40 , respectively. The y-axis represent accuracy. The black dots on each ratio represent all ten accuracy from 10 trials. The blue plot is the averaged accuracy of ten trials for each ratio. Curve-fitting was done for the averaged accuracy, and it is shown in the red dashed curve. The accuracy in reduction 1 dropped exponentially as ratio decreased, following the curve fitted equation 8. The coefficient of determination (R^2), was 0.9752.

$$\text{accuracy} = -0.6846e^{-28.48 \times \text{ratio of reduction}} + 0.9996 \quad (8)$$

Although the exponential accuracy drop, the averaged accuracy remained near 90%. We could see here that prediction of airfoil kinematics could be done with high accuracy even in substantial reduction of window size. The averaged accuracy and corresponding standard deviation is shown in Table. 2.

2. Reduction 2. Downampling

The result of reduction 2 is shown in Fig. 8. The x-axis represents the final image size in ratio. This means that a base window ratio of $1/3$ with strides = 1 will be in category $1/6$ label, a base window of $1/4$ with strides = 2 will be in category $1/12$ label, and so on. The colors of the bar graphs represent the base window ratio, and the y-axis represent the accuracy. The black plot with an 'o' shaped marker is the result from reduction 1, inserted for comparison between reduction 1 and reduction 2. The black I-shaped line on each bar represent the error range of each bar, on 10 different training runs. The height of the bars shows the averaged accuracy. The values of the plot are listed in table 3.

Table 2 The averaged accuracy and its standard deviation of reduction 1.

| Ratio # | Averaged Accuracy(%) | Standard Deviation(%) |
|---------|----------------------|-----------------------|
| 1/3 | 100 | 0 |
| 1/4 | 100 | 0 |
| 1/5 | 98.923 | 0.3889 |
| 1/6 | 99.500 | 0.0476 |
| 1/7 | 99.256 | 0.2705 |
| 1/8 | 98.357 | 0.6948 |
| 1/9 | 97.274 | 1.0264 |
| 1/10 | 96.060 | 1.6359 |
| 1/11 | 94.440 | 1.9587 |
| 1/12 | 93.202 | 0.6104 |
| 1/13 | 92.798 | 0.9241 |
| 1/14 | 89.792 | 1.8725 |
| 1/15 | 90.631 | 1.1576 |

Table 3 The averaged accuracy and its standard deviation for all ratios of reduction

| Final Image Size | Base Ratio | Strides | Averaged Accuracy (%) | Standard Deviation (%) |
|------------------|------------|---------|-----------------------|------------------------|
| 1/6 | 1/3 | 1 | 100 | 0 |
| 1/8 | 1/4 | 1 | 99.982 | 0.054 |
| 1/9 | 1/3 | 2 | 100 | 0 |
| 1/10 | 1/5 | 1 | 99.708 | 0.58 |
| 1/12 | 1/3 | 3 | 100 | 0 |
| 1/12 | 1/4 | 2 | 99.929 | 0.143 |
| 1/12 | 1/6 | 1 | 99.637 | 0.193 |
| 1/14 | 1/7 | 1 | 98.458 | 0.649 |
| 1/15 | 1/3 | 4 | 100 | 0 |
| 1/15 | 1/5 | 2 | 99.923 | 0.088 |

The performance of airfoil kinematics prediction from reduction 2, downsampling, was extremely high. The lowest case was averaged accuracy of 98.458% (base ratio = 1/7, strides = 1). The model performance was nearly perfect, approaching nearly 100% for all tested cases.

We found that base window size, or measuring field area, governed the prediction performance of CNN. Having a larger base ratio performed better than that of a smaller base ratio, regardless of the final image size. For example, base ratio = 1/3 (blue bar) always performed better than that of base ratio 1/4, 1/5, 1/6, and 1/7, the averaged accuracy of base ratio = 1/4 (orange bar) always performed better than that of 1/5, 1/6, and 1/7, the averaged accuracy of base ratio = 1/5 (green bar) always perform better than that of smaller base ratio, and so on. The accuracy difference from strides size was extremely small. The maximum accuracy difference caused by strides was 0.215%, for base ratio = 1/5 between strides = 1 and 2.

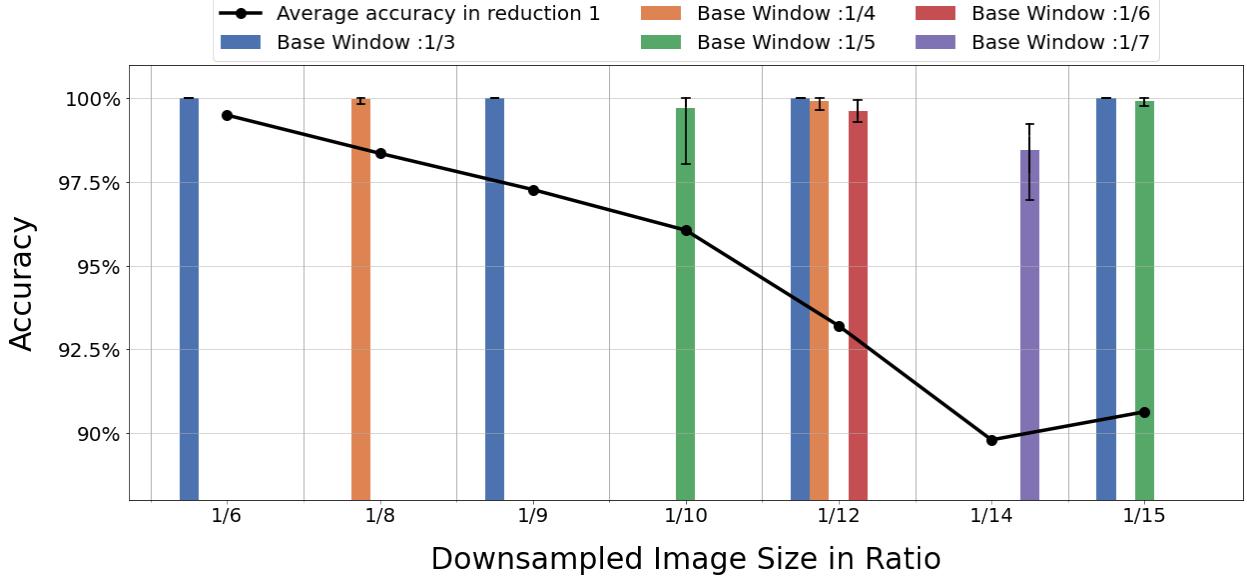


Fig. 8 The result for the reduction 2.

B. Classification with Sparse Basis Selection

An alternative classification methods, sparse basis selection, was tested as a benchmark. The result of this method in comparison with CNN model of reduction 1 is shown in table 4. We can note here that the CNN models, are performing with much higher accuracy. Especially when decreasing the measuring area, or the window size, the sparse basis selection model are influenced significantly more than the CNN models. This shows that sparse basis selection method will not be feasible for applications that can only get measurement from a small window, or applications that require significantly high accuracy.

Table 4 The prediction accuracy comparison between CNN model and sparse basis selection method

| Ratio | CNN Model Accuracy(%) | Sparse Basis Selection Accuracy(%) |
|-------|-----------------------|------------------------------------|
| 1/3 | 100 | 88.571 |
| 1/6 | 99.500 | 60.714 |
| 1/9 | 97.274 | 45.060 |
| 1/12 | 93.202 | 26.726 |
| 1/15 | 90.631 | 28.869 |

Also, there was a trade-off between training time and predicting time. For CNN models, it took significant time to train the model; while the prediction time was relatively short. The sparse basis selection method had opposite in time consumption. It took relatively short time to create library for all the basis, but it took much longer time for prediction. This is largely due to the nature of sparse basis selection prediction method, which solves an optimization problem to categorize a given matrix. This could be detrimental to a system which require a fast feedback for control.

However, the sparse basis selection model has its strength in predicting with sparsely-sensed measurement. While

CNN models can perform with inputs that are significantly smaller measuring field area compared to sparse basis selection, the model is infeasible if there are only few measurements, e.g. 50 measuring points. In Ref. [11], the prediction of airfoil pitching kinematics was done using only 50 measuring points that have been randomly selected, scoring overall accuracy of 78%. Considering the significant reduction of input measuring points, sparse basis selection model clearly has its strength and suitable applications.

C. Chirp Signal Pitching Airfoil Kinematics Prediction

1. Test 1 - Evaluation of Chirp Signal on Model Trained from Sinusoidal Pitching Data

The prediction of chirp signal pitching kinematics with $t_{end} = 9000$ was tested on a model trained from sinusoidal pitching of ratio 1/3. The anticipation was that CNN model would be capable of identifying the underlying pattern in the data, specifically looking for patterns in the same base angle of attack and frequency. However, despite this expectation, the accuracy of the CNN in handling this data was found to be extremely low. The average accuracy was 27.4% with a standard deviation of 0.397%. The confusion matrix in Fig. 9 also shows that the model was not able to estimate the kinematics, having no trend of prediction in diagonal entries.

This was tested for a more strictly categorized frequency as the low accuracy may be coming from the large variance of continuous frequency values being categorized into the same category. We allowed only 0.01 dimensionless frequency difference between categorized frequency and continuous frequency. The average accuracy dropped to 17.9%, confirming that categorization was not the problem.

2. Test 2 - Evaluation of Chirp Signal on Model Trained from Sinusoidal + Chirp Data

Seeing that the prediction of the chirp signal couldn't be done on models trained on sinusoidal data, a new model was trained using sinusoidal data and chirp signal of $t_{end} = 9000$ together. New model was able to predict chirp signal pitching kinematics with high accuracy while maintaining the sinusoidal pitching kinematic prediction near perfect. The average accuracy for sinusoidal pitching case was 99.8%, with a standard deviation of 0.168%. For the chirp signal, the average accuracy was 95.5%, with a standard deviation of 2.93%. Fig. 10 shows the confusion matrix.

3. Test 3 - Evaluation of Out-of-Distribution Chirp Signal on Model Trained from Sinusoidal + Chirp Data

Encouraged by success in test 2, we took a step further to test if the model from test 2 can predict chirp signal with $t_{end} = 6000$ (frequency changing faster). We were anticipating that training with chirp signal with $t_{end} = 9000$ would implement the effect of increasing frequency, making the model to be able to characterize the kinematics of faster-increasing chirp signal($t_{end} = 6000$). The average accuracy in test 3 was 29.4% and the standard deviation was 2.72%. This was frustrating to see that the model, even though implemented with varying frequency, couldn't predict the airfoil kinematics when the rate of change was slightly different. However, checking the confusion matrix 11, the

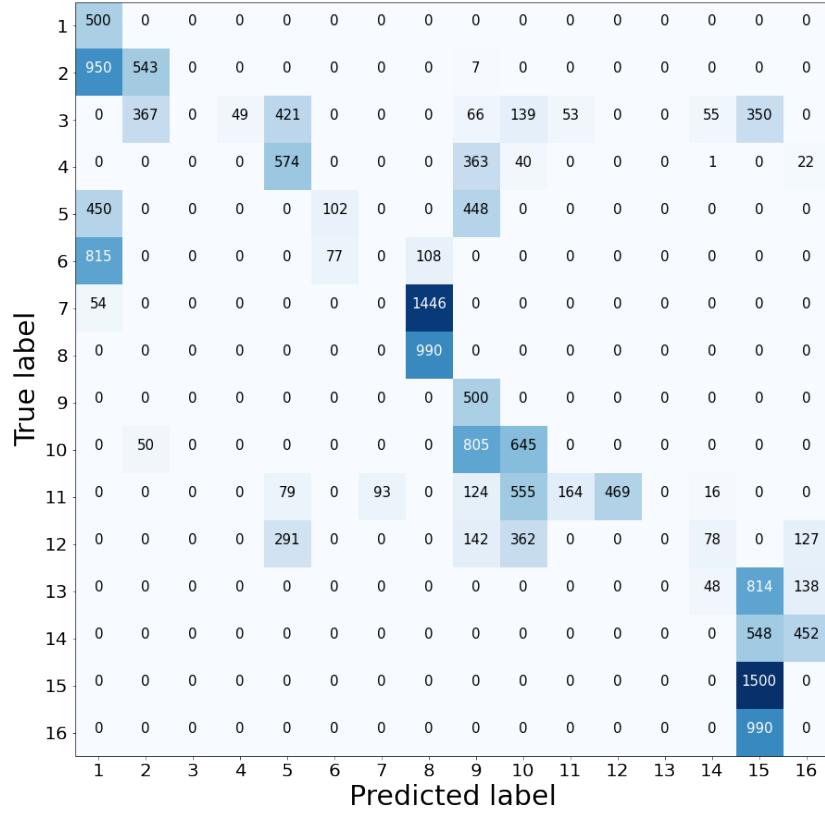


Fig. 9 Confusion matrix for the prediction of chirp signal on a model trained from sinusoidal pitching data. Class indices of 1-16 correspond to $(\alpha_0, f_p) = (25^\circ, 0.05), (25^\circ, 0.1), \dots, (25^\circ, 0.5), (30^\circ, 0.05), \dots, (30^\circ, 0.5)$, respectively.

prediction had a clear trend of staying near the diagonal line. This gave the possibility for prediction of chirp signal of different rates of change when implemented with the varying frequency in training if we could build a CNN model that was more robust.

VII. Conclusion

In this article, we explored the estimation of airfoil pitching kinematics using convolutional neural networks. The estimation was done from the instantaneous velocity field snapshot downstream of the wake. Two types of pitching kinematics were subjected to prediction: sinusoidal pitching and chirp signal pitching.

For the sinusoidal pitching, we tested on 16 categories. The categories consisted of combination of two base angles of attack, $(\alpha_0 = 25^\circ, 30^\circ)$ and eight dimensionless pitching frequency ($f_p = 0.05, 0.1, 0.2, 0.25, 0.3, 0.35, 0.4, 0.5$). The pitching amplitude was set constant to 5° . The CNN was challenged by reducing the input image size in two ways:

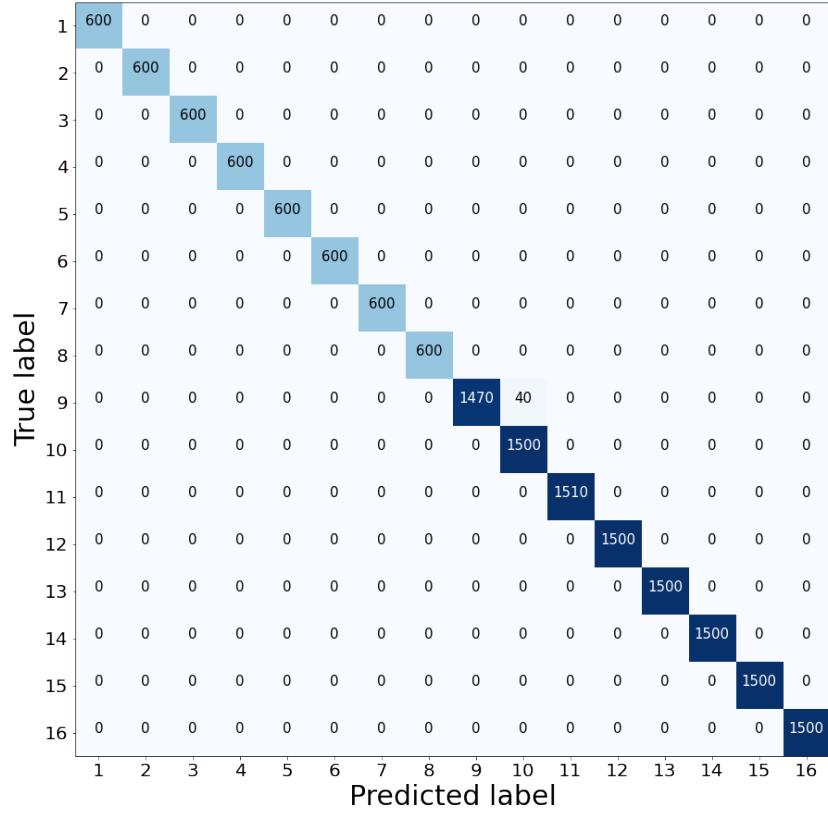


Fig. 10 Confusion matrix for the prediction of sinusoidal signal on a model trained from sinusoidal + chirp data. Class indices of 1-16 correspond to $\alpha_0, f_p = (25^\circ, 0.05), (25^\circ, 0.1), \dots, (25^\circ, 0.5), (30^\circ, 0.05), \dots, (30^\circ, 0.5)$, respectively.

reduction 1. reducing the measuring field area, and reduction 2. downsampling the input image. In both reductions, the airfoil pitching kinematics could be estimated with high accuracy even with substantial reduction.

In reduction 1, the accuracy dropped exponentially as the input image size got smaller. Although the accuracy dropped, the lowest average accuracy was 89.792%. The exponential drop followed Eq. 8. The results are plotted in Fig. 7, and listed in Table. 2.

Reduction 2 showed much higher performance compared to reduction 1. The lowest average accuracy was 98.458% for reduction 2. We also found that the measuring field area was dominant for the performance of CNNs. Having a larger base window size with bigger strides performed better even if the final input image size was smaller when compared with a smaller base window size with smaller strides having a larger final input image size. The difference in accuracy caused by downsampling for the same base window size was almost negligible compared to the effect of base

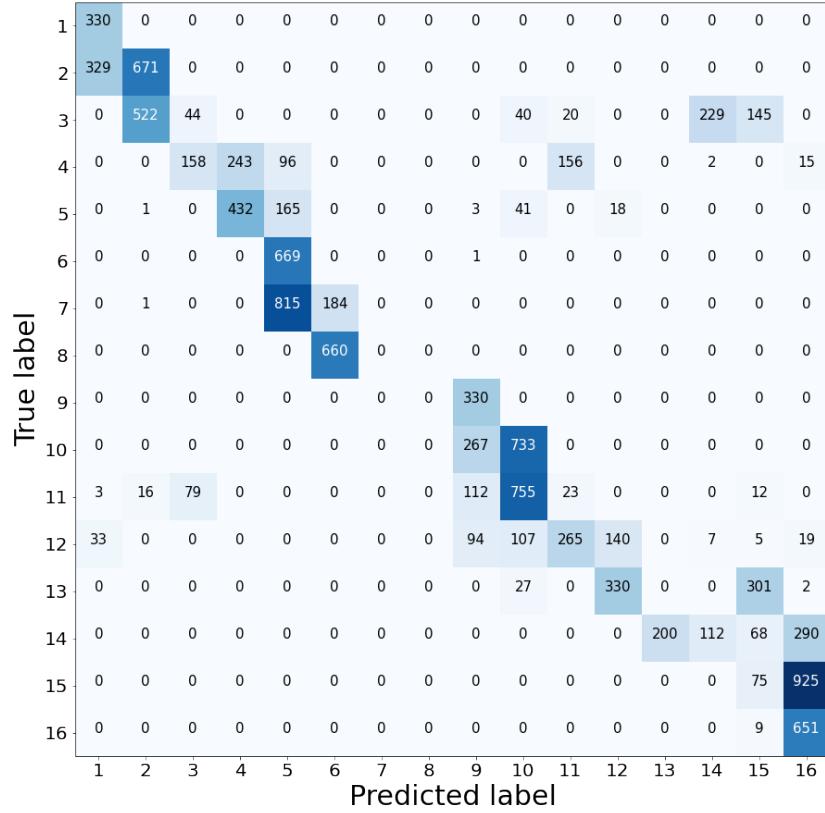


Fig. 11 Confusion matrix for the prediction of faster-changing chirp signal on a model trained from sinusoidal + chirp data. Class indices of 1-16 correspond to $(\alpha_0, f_p) = (25^\circ, 0.05), (25^\circ, 0.1), \dots, (25^\circ, 0.5), (30^\circ, 0.05), \dots, (30^\circ, 0.5)$, respectively.

window size. The results are shown in Fig. 8.

Looking at results from the two types of reduction in sinusoidal pitching, downsampling (reduction 2) seemed to be worth more examination to test its limit in size of strides to maintain favorable accuracy. However, reducing the input image size has its limit when applied to CNNs. To reduce the input image size beyond a certain level, the depth of CNN architecture must be reduced as well. This is because CNNs work by decreasing the dimensionality of the input image layer by layer. Using the same deep CNNs in smaller input image size results in a negative dimension, and thus is not feasible.

This raises the question of the tradeoff between downsampling and model depth. Assuming that the measuring field area is sufficient, downsampling the image from the field will have a benefit in the cost of prediction. How far can we sparsely sample the velocity along with the reduction of CNNs's depth while maintaining a reasonable range of

accuracy? This is the next step for the idea to possibly expand. Also, even the reduction of window size (reduction 1), although seemed less significant compared to downsampling, was performing nearly over 90%, meaning that the model needs to be challenged more in the same sense.

The performance of CNN and sparse basis selection method was compared, as a benchmark. As shown in Table. 4, the accuracy of CNN was significantly higher than that of sparse basis selection. Also, the adversarial effect of reduction of window size impacted sparse basis selection model much more significantly. The prediction time for sparse basis selection method was much longer compared to the CNN models, while training time being much shorter. Showing that sparse basis selection method may not be suitable for applications that require fast prediction. However, the sparse basis selection can work with only few measuring point data given while the CNN models need much more input data point. Sparse basis selection may have advantages in situations involving only a few sensing points in a wider measuring field area compared to CNN.

Looking at results from the two types of reduction in sinusoidal pitching, downsampling (reduction 2) seemed to be worth more examination to test its limit in size of strides to maintain favorable accuracy. However, reducing the input image size has its limit when applied to CNNs. To reduce the input image size beyond a certain level, the depth of CNN architecture must be reduced as well, as CNNs work by decreasing the dimensionality of the input image layer by layer. Using the same deep CNNs in smaller input image size will result in a negative dimension, and thus is not feasible. This raises the question of the tradeoff between downsampling, or sparse sensing and model depth. Assuming that the measuring field area is sufficient, sparse sensing from the field will have a benefit in the cost of prediction. How far can we sparsely sample the velocity along with the reduction of CNNs's depth while maintaining a reasonable range of accuracy? This is the next step for the idea to possibly expand. Also, even the reduction of window size (reduction 1), although seemed less significant compared to downsampling, was performing nearly over 90%, meaning that the model needs to be challenged more in the same sense.

For the chirp signal pitching, we increased the pitching frequency linearly. There were two different rates of change for the frequency: $t_{end} = 9000$ and $t_{end} = 6000$. $t_{end} = 9000$ was relatively slowly increasing frequency and $t_{end} = 6000$ had relatively faster increase. The pitching motion is described in 2 3. We tried three tests on the estimation of chirp signal pitching motion. The first test evaluated the estimation of chirp signal kinematics of $t_{end} = 9000$ on a model that was trained on sinusoidal pitching. The window size was 200x200, which corresponds to the 1/3 ratio for the reduction of measuring field area. This was tested as we were anticipating that the model could perform proper wake classification from the fundamental difference in the base angle of attack and instantaneous frequency although the varying frequency can create big changes in the wake. However, the test result was an average accuracy of 27.4%, showing no remarkable performance in the estimation of pitching kinematics. The varying frequency seemed to be a clear out-of-distribution data for the model, as there was no clear trend of prediction being near the diagonal line in confusion matrix 9. Even when the continuous frequency was more strictly classified, the model was not able to

estimate the kinematics, confirming that the classification of frequency was not the reason for the low performance.

The model trained on the sinusoidal data could not spot the fundamental similarity between sinusoidal pitching and chirp signal pitching. So, we moved on to the second test to check if the model could estimate the airfoil pitching kinematics when the model had a chirp signal in its training data. We used a mixture of sinusoidal pitching and chirp signal pitching with $t_{end} = 9000$, which was in a ratio of 6:1. This imbalance was intended to maintain the high performance on sinusoidal pitching while informing the model with a glimpse of chirp signal cases. This attempt was successful, recording an average accuracy for chirp signal pitching of 95.5%, and an average accuracy for sinusoidal pitching of 99.8%. The two confusion matrices are given here: ?? 10. The result is very favorable as the accuracy for sinusoidal pitching remained in the region near perfect, while the chirp signal prediction improved significantly.

Having known that the model could learn the effect of varying frequency, we tested on a more rapidly changing chirp-signal-pitching, with $t_{end} = 6000$ (1.5 times more quickly increasing compared to $t_{end} = 9000$). The model could not perform well anymore, and the average accuracy was 29.4% in this case. From the confusion matrix11, however, we could still see that the prediction stayed near the diagonal line of correct prediction. The model learned the chirp-signal pitching pattern in some way but lacked in generalizing the patterns. To check if the error from the categorization of continuous frequency is causing this error, we also tried for strict categorized data, with frequency error under 0.01, and the averaged accuracy was 25.9%. This proved that the categorization was not the problem. Here is the confusion matrix: ??.

Combining the result from the sinusoidal pitching and non-sinusoidal pitching, CNN models could perform nearly perfectly when provided the exact case in the training process. In CNN models that could perform perfectly, the input image size could even be significantly reduced, and still performed with high accuracy. This is promising as the basic concept of using an instantaneous snapshot of the velocity field in the wake far from the airfoil for the estimation of the pitching kinematics is feasible. However, the biggest problem with the CNN architecture that we used was the lack of robustness. This is not good for practical purposes, where we can frequently encounter out-of-distribution data. Also, the reduction of input image size in sinusoidal pitching inspired us to test the reduction of image size along with the reduction of network depth, where shallower models are expected to have more robustness in prediction. This motivates us to the next step of estimating airfoil kinematics from a snapshot of the velocity field, to create a more robust model, with less depth, or possibly completely different ways.

References

- [1] Ravindran, S. S., “A reduced-order approach for optimal control of fluids using proper orthogonal decomposition,” *International journal for numerical methods in fluids*, Vol. 34, No. 5, 2000, pp. 425–448.
- [2] Sirovich, L., “Turbulence and the dynamics of coherent structures. I: Coherent structures. II: Symmetries and transformations. III: Dynamics and scaling,” *Quarterly of applied mathematics*, Vol. 45, No. 3, 1987, pp. 561–590.

- [3] Schmid, P. J., “Dynamic mode decomposition of numerical and experimental data,” *Journal of fluid mechanics*, Vol. 656, No. August, 2010, pp. 5–28.
- [4] Rowley, C. W., Mezić, I., Bagheri, S., Schlatter, P., and Henningson, D. S., “Spectral analysis of nonlinear flows,” *Journal of fluid mechanics*, Vol. 641, 2009, pp. 115–127.
- [5] Eivazi, H., Veisi, H., Naderi, M. H., and Esfahanian, V., “Deep neural networks for nonlinear model order reduction of unsteady flows,” *Physics of Fluids*, Vol. 32, No. 10, 2020.
- [6] Maulik, R., Lusch, B., and Balaprakash, P., “Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders,” *Physics of Fluids*, Vol. 33, No. 3, 2021.
- [7] Viquerat, J., and Hachem, E., “A supervised neural network for drag prediction of arbitrary 2D shapes in laminar flows at low Reynolds number,” *Computers & Fluids*, Vol. 210, 2020, p. 104645.
- [8] Rocha Ribeiro, B. L., and Franck, J. A., “A machine learning approach to classify kinematics and vortex wake modes of oscillating foils,” *AIAA AVIATION 2021 FORUM*, 2021, p. 2947.
- [9] Yoda, M., “Super-Resolution Imaging in Fluid Mechanics Using New Illumination Approaches,” *Annual Review of Fluid Mechanics*, Vol. 52, No. 1, 2020, pp. 369–393. <https://doi.org/10.1146/annurev-fluid-010719-060059>, URL <https://doi.org/10.1146/annurev-fluid-010719-060059>.
- [10] Yang, G., and Sommer, S., “A Denoising Diffusion Model for Fluid Field Prediction,” *arXiv.org*, 2023.
- [11] Towne, A., Dawson, S. T. M., Brès, G. A., Lozano-Durán, A., Saxton-Fox, T., Parthasarathy, A., Jones, A. R., Biler, H., Yeh, C.-A., Patel, H. D., and Taira, K., “A Database for Reduced-Complexity Modeling of Fluid Flows,” *arXiv.org*, 2022.
- [12] Bright, I., Lin, G., and Kutz, J. N., “Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements,” *Physics of Fluids*, Vol. 25, No. 12, 2013.
- [13] Bright, I., Lin, G., and Kutz, J. N., “Classification of spatiotemporal data via asynchronous sparse sampling: Application to flow around a cylinder,” *Multiscale Modeling & Simulation*, Vol. 14, No. 2, 2016, pp. 823–838.
- [14] Brunton, S. L., and Kutz, J. N., *Data-driven science and engineering: Machine learning, dynamical systems, and control*, Cambridge University Press, 2019.