# Application of Convolutional Neural Networks for Feature Identification in Complex Fluid Flows

Shizheng Wen[1]
*Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China*

Michael W. Lee,[2] Kai M. Kruger Bastos,[3] and Earl H. Dowell[4]
*Duke University, Durham, NC, 27708, USA*

For Peer Review

**Recent efforts have shown machine learning to be useful for analysis of nonlinear fluid dynamics. Predictive accuracy is often a central motivation for employing neural networks, but the pattern recognition central to the network's function is equally valuable for purposes of enhancing our dynamical insight into sometimes confounding dynamics. In this paper, convolutional neural networks (CNNs) were trained to recognize several qualitatively different subsonic buffet flows over a high-incidence airfoil. The convolutional kernels and corresponding feature maps, developed by the model with no temporal information provided, identified large-scale coherent structures in agreement with those known to be associated with buffet flows. Sensitivity to hyperparameters including network architecture and convolutional kernel size was explored. One conclusion is that only a small training dataset is necessary, but that smaller kernels are better at coherent structure identification than are larger kernels. A long-short term memory CNN was then used to demonstrate that with the inclusion of temporal information, the coherent structures remained qualitatively comparable to those of the conventional CNN and less dynamically significant features were no longer recorded. The coherent structures identified by these models enhance our dynamical understanding of subsonic buffet over high-incidence airfoils over a wide range of Reynolds numbers.**

[1] Undergraduate researcher, School of Energy and Power Engineering
[2] PhD candidate, Department of Mechanical Engineering and Materials Science, AIAA Student Member
[3] PhD candidate, Department of Mechanical Engineering and Materials Science
[4] Professor, Department of Mechanical Engineering and Materials Science, AIAA Honorary Fellow

Submitted to AIAA Journal. Confidential - Do not distribute.

## I. Introduction

IN recent years, with the development of high-performance computing architectures and experimental measurement capabilities [1-3], fluid researchers are able to obtain high precision and high-resolution spatiotemporal data from large-scale fluid computational simulations and experiments. Too, the advancement of sophisticated algorithms and the abundance of open source software enables researchers to apply machine learning (ML) to address many challenges. Turbulence modeling and, more generally, nonlinear fluid dynamics has been one proving ground for neural networks.

For nonlinear fluid flow regimes, incorporating domain knowledge into learning algorithms has been demonstrated to be feasible [4]. Data-driven turbulence modeling presents promising extensions to more conventional numerical system closure techniques and are therefore of significant value for engineering applications [5]. For example, Tracey et al. [6] successfully reproduced the Spalart-Allmaras turbulence model by replacing the deliberately removed source term with machine-learned functional forms. Duraisamy’s group also pursued efforts in data-driven turbulence modeling [7-11] with encouraging results; they were able to infer functional forms of modeling discrepancies by using inverse modeling, and then were able to reconstruct the patterns with ML for incorporation into turbulence model source terms. Xiao’s group emphasized the physical constraint of Reynolds stress and proposed the concept of a physics-informed machine learning approach [12, 13]. Ling et al. [14] reconstructed the mapping relations between field variables and the Reynolds stress anisotropy tensor, and replaced the turbulence model with a tensor-based neural network. Zhu et al. [15] completely replaced the Reynolds stress transport equations with neural networks and then constructed a mapping function between the turbulent eddy viscosity and the mean flow variables.

In addition to data-driven turbulence modeling, characteristics of deep learning (DL) algorithms [16-18] like the convolutional neural network (CNN) and the long-short term memory network (LSTM) provide methods for researchers to evaluate the temporal and spatial patterns in data. For example, Zhang et al [19] trained multiple linked CNNs to learn the lift coefficients of an airfoil with a variety of shapes in multiple flow regimes. Guo et al. [20] proposed a convolutional encoder-decoder approach that can predict steady velocity and pressure fields. Bhatnagar et al. [21] improved the computational efficiency of this effort by sharing encoder-decoder layers. Mohan et al. [22] built a deep learning approach to reduced order modeling (ROM) for isotropic turbulent flows by replacing Galerkin projection with LSTM neural networks.

For Peer Review

Submitted to AIAA Journal. Confidential - Do not distribute.

The success of the aforementioned works indicates the encouraging prospects of ML in fluid mechanics. Furthermore, it also adequately demonstrates that ML can extract intrinsic flow features for use in establishing a nonlinear mapping relationship with the desired output. However, previous research efforts have focused primarily on the accuracy of predictive variables, without studying closely the information hidden inside the learning model itself. Additionally, the more dynamically motivated efforts which do exist have utilized flows with high levels of symmetry, e.g. isotropic turbulence. In this paper, we employ conventional ML implementations to identify coherent structures in a flow important to engineering applications, i.e. one over an airfoil at a high angle of attack, where subsonic buffet is known to occur. The coherent structures associated with buffet, identified through feature maps associated with the convolutional kernels, align with and expand upon the previous mathematical and physical insights for the problem despite being identified entirely by the ML algorithms. A side-effect of this coherent structure identification was a near-perfect flow identification capability, where the neural network learned quickly how to recognize qualitatively different flow regimes from individual temporal snapshots.

The remainder of this paper is organized as follows. In section II, the problem of subsonic buffet and the computation of the flow data provided to the neural networks are discussed. In section III, results from the CNN architecture are presented and sensitivity to certain hyperparameters are discussed. In section IV, results from the LSTM architecture are presented with the conclusion that little difference is observed between the CNN and LSTM coherent structures.

For Peer Review

## II. Problem Formulation

Flow over a NACA 0012 airfoil was computed via unresolved direct numerical simulation (UDNS) at a constant incidence angle of 40 degrees and temporally constant Reynolds numbers ranging from 100 to 1,000,000. The Mach number remained constant in all flows at approximately 0.05. These conditions elicited a spectrum of temporally fluctuating flows over the airfoil: some periodic, some quasi-periodic, and some chaotic. These unsteady, lift-generating flows over the airfoil are collectively known as subsonic buffet.

### A. Subsonic Buffet

The buffet phenomenon in subsonic open flows has received recent attention [23, 24]. Vortex shedding similar to bluff body flows is observed in high-incidence flows around stationary airfoils [23-25]. Lift coefficients are one way to characterize these time-periodic dynamic instabilities. A scaling analysis of the Navier-Stokes equations [26]

Submitted to AIAA Journal. Confidential - Do not distribute.

demonstrates that both the peak-to-peak oscillating lift and reduced frequency in buffeting flows are of order unity. As such, while the kinematics of the flow are not yet fully understood, the buffet dynamics are of large-enough scale to dramatically affect the airfoil performance. It is these large-scale structures that are identified through ML in this research.

## B. UDNS Flow Simulations

The neural networks central to this research were presented with snapshots of fluid flows around an airfoil across a range of Reynolds numbers. The subsonic flows exhibited characteristics of buffet, in which the airfoil's lift coefficient oscillated in time but had a positive mean value. The method of simulating such a flow is an active area of research; readers are directed to [28] in the submitted manuscript for further reading. URANS simulations have been shown to present reduced frequencies comparable to those of experiments, but the amplitudes of the oscillations were more sensitive to the choice of closure model. The URANS simulations also failed to exhibit buffet at all at some lower Reynolds numbers which were known from experiments to yield buffet.

Recent efforts utilizing "unresolved direct numerical simulation" (UDNS), also known as "full Navier-Stokes" (Full NS), have provided a numerically tractable method of simulating such flows with large-scale behavior comparable to that of experiments. In particular, the reduced frequencies of the lift and drag coefficients were computed to be within 7% of the experimental results at the same high Reynolds numbers [28].

This method is known to be only conditionally stable in time. Too, it is understood that this method does not guarantee, nor does it seek, accuracy in subgrid-scale dynamics. However, for unsteady aerodynamic phenomena dominated by large-scale flow structures, e.g. subsonic buffet, UDNS presents a compromise between computational cost and agreement with experiment. It was thus utilized in this study to generate a snapshot database for use by the neural networks, which themselves only studied large-scale flow structures. The pattern recognition of the neural network was not contingent upon fully resolved small-scale flow features, and thus the purpose of this study did not require resolution down to the Kolmogorov scale.

Flows were thus simulated in time about a symmetric NACA 0012 airfoil at several Reynolds numbers: 100, 600, 1,000, 10,000, 100,000 and 1,000,000. All six simulations started with a control volume at rest and converged to a statistically steady state; a constant time step of 0.002 seconds was employed. The incidence angle was held at 40 degrees and Mach number was held constant at approximately 0.05: strongly within the incompressible limit.

For Peer Review

Simulations were performed with a truncated NASA grid [27], with 257 airfoil surface points, as illustrated in Fig. 1. The grid extends 20 chord lengths in all directions, which was determined to be adequate to resolve far-field behavior [28]. The Navier-Stokes equations were solved directly in the ANSYS Fluent software package to generate all flow data used in this study. At a Reynolds number of 87,000, at which the flow parameters matched those of Tang and Dowell's [23] experimental study, the lift and drag coefficient reduced-frequencies were observed to be in good agreement between the UDNS simulation and experimental results.

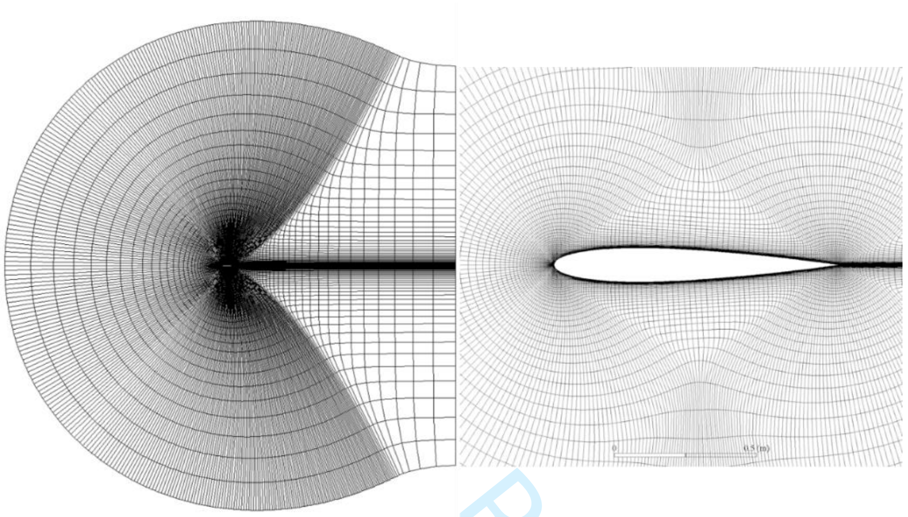


**Fig. 1 Full view (left) and close-up view (right) of the 2D airfoil grid used for UDNS simulations.**

Review

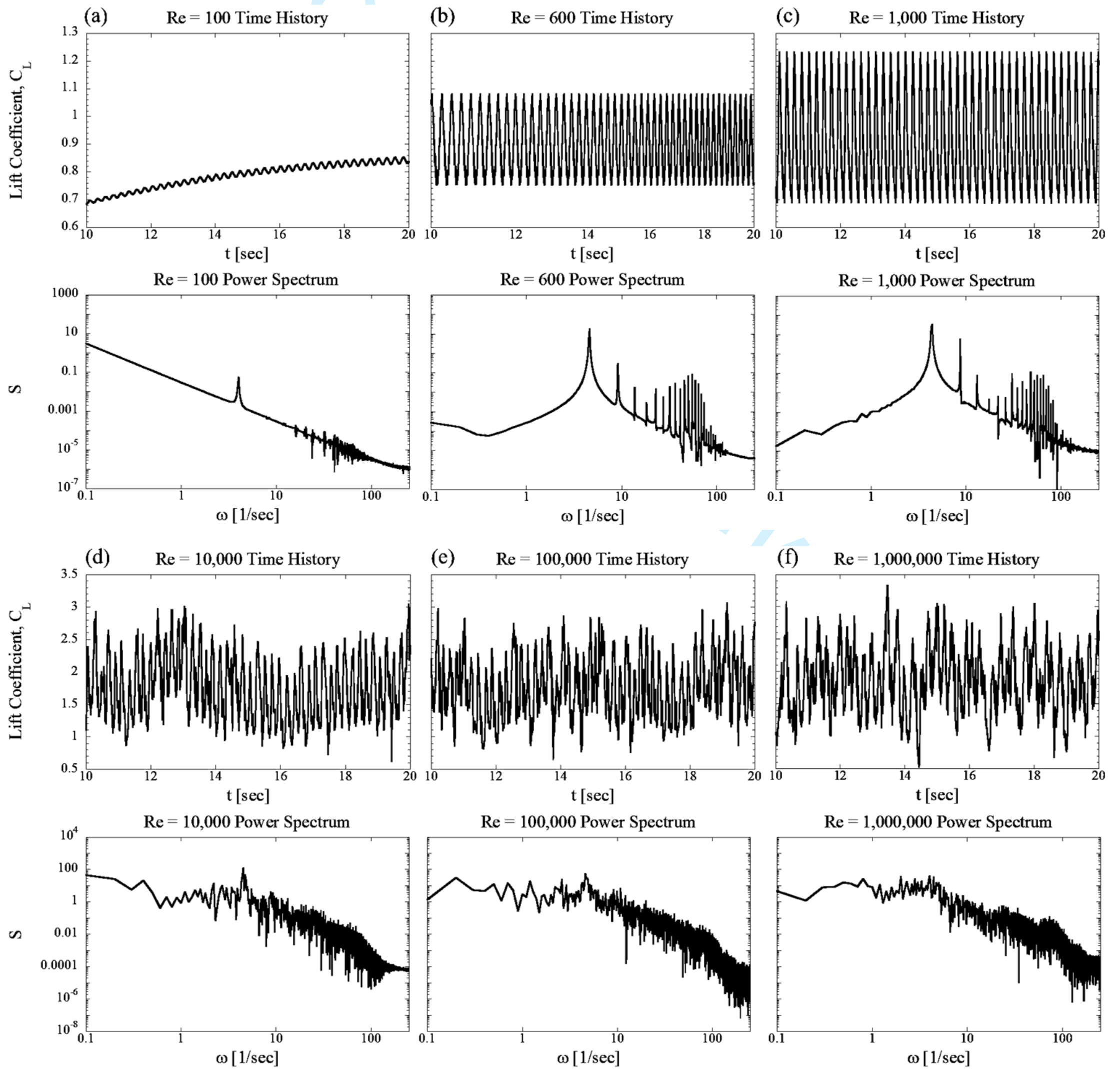
Figure 2 shows steady-state time histories and power spectra of the lift coefficient (CL) as computed at different Reynolds numbers. By employing the heuristic flow characterization developed by Wiebe and Virgin [29] and in agreement with existing literature [30], each Reynolds number is associated with qualitatively different flow regimes: periodic flow, quasi-periodic flow and chaotic flow, as shown in Table 1. All simulations were performed with a time step of 0.002 seconds. As such, the 2,000 snapshots at Re = 100, for example, correspond to the flow as simulated between 16 and 20 dimensional seconds after the simulation was started from rest.

Recent work by one of the authors [28] determined that there was weak dependence on grid resolution for large ranges of Reynolds number when seeking the lift and drag coefficient reduced frequencies. As such, the reduced frequencies are assumed to be dominated by large-scale features. It is these local but large-scale flow features that this study has identified by using ML.

Submitted to AIAA Journal. Confidential - Do not distribute.

**Table 1 High-fidelity UDNS snapshots provided to the neural networks, classified by Reynolds number as qualitatively different flow regimes.**

| Reynolds Number | Sampling region $(t_s - t_f)/\Delta t$ | Flow regime |
|---|---|---|
| 100 | 8000 – 10,000 | Periodic flow |
| 600 | 8000 – 10,000 | Quasi-periodic flow |
| 1,000 | 8000 – 10,000 | Quasi-periodic flow |
| 10,000 | 5000 – 10,000 | Chaotic flow |
| 100,000 | 5000 – 10,000 | Chaotic flow |
| 1,000,000 | 5000 – 10,000 | Chaotic flow |



Submitted to AIAA Journal. Confidential - Do not distribute.

**Fig. 2 Response of lift coefficient changing with time and corresponding power spectrum at Reynolds numbers of 100, 600, 1,000, 10000, 100,000 and 1,000,000.**

### C. Data Preprocessing

Because the lift coefficient is only determined by the airfoil surface pressures and the near-wall flow dynamics are indicative of the qualitative flow behavior, the flow region for ML consideration was confined to the area shown in Fig. 3(a). It is clear that the UDNS grid structure is not Cartesian. For easier application of a square Cartesian convolutional kernel to the flow snapshots, the spatial grid was thus adjusted to be Cartesian. The subsequent 200-by-150 pixel Cartesian grid is shown in Fig. 3(b). Flow values within the airfoil (the blue area in Fig. 3(b)) area were set to zero. Splines defined by the NACA airfoil standard [31] were used to determine what points were within the airfoil area.

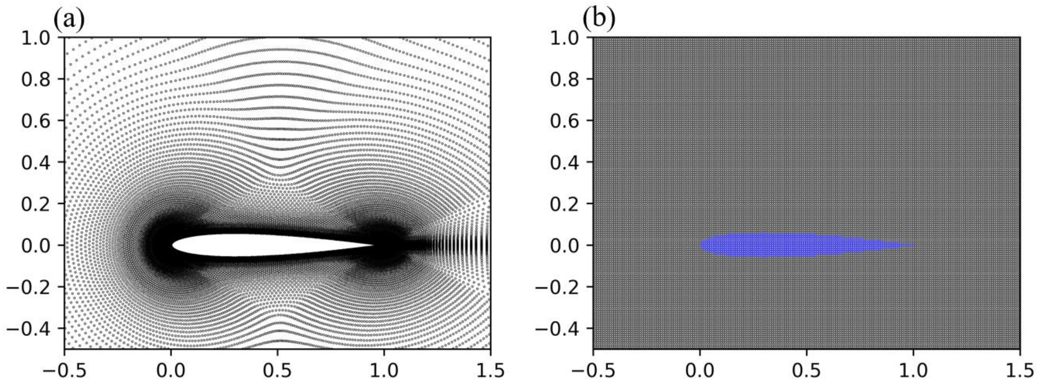


**Fig. 3 UDNS grid subsection (a) and the Cartesian grid (b) provided to the neural networks.**

## III. Convolutional Neural Network Implementation

A convolutional neural network was employed to identify the large-scale coherent structures associated with the qualitatively different manifestations of buffet. All neural networks in this paper were constructed within the Google tensorflow [41] environment. A summary of the mathematics of the CNN is presented in the appendix. The trained model was able to virtually unilaterally discern the qualitative flow state based on a single arbitrary temporal snapshot. Too, the model's kernels – depending somewhat on the kernel size – were able to identify all dynamically significant large-scale flow features with no physical insight provided a priori.

### A. Formulation

Submitted to AIAA Journal. Confidential - Do not distribute.

The artificial neural networks (ANNs) first became prominent in the field of artificial intelligence in the 1980s [32]. They are straightforward abstractions of biological neurons, realized as elements in an artificial network like a program or silicon-based circuits [33]. They present advantages in learning nonlinear and complex relationships, good generalization of unseen data and simplicity of input data formatting. For these and other reasons, ANNs continue to play an important role in modern ML methods. However, due to the fully-connected nature of neurons in two adjacent layers, traditional ANNs will generate a large number of parameters – many of which will likely be superfluous – when adding additional layers or increasing the number of neurons in each layer. This in turn often gives rise to overfitting of the model as the optimization procedure converges. With multidimensional data, even something as simple as static images, interpreting them as 1D vectors (as is necessary in traditional ANNs) will 1) cause an explosion of parameters and 2) destroy the spatial relationships which define the multidimensional inputs. To address these limitations, a specific ANN architecture was developed: the convolutional neural network (CNN). LeCun et al. [34] firstly proposed the layout of LeNet-5 in 1988 and was notably successful in a handwritten digit recognition task. Subsequently, many architectures derivative of the LeNet-5 further explored the image recognition problem, with much success [35-37].

CNNs derive their utility from seeking patterns at multiple scales through the process of convolving small kernels – perhaps five pixels square – with a much larger input signal – perhaps a grayscale image of 100 pixels square. CNNs are particularly successful when the input data can be decomposed into some form of hierarchical basis representation; and this is often labeled as automatic feature extraction [38]. The optimization problem stems from finding kernels which identify the most fundamental patterns in the signal, which can then combine at higher levels to yield larger-scale patterns which in turn reconcile with provided data labels. For example, a face may be identified in an image at a high layer of the CNN because eyes and mouths are identified at a lower CNN layer, which are themselves identified at the lowest layer by kernels which design themselves to recognize edges and angles in the provided image. While the problem remains one of optimizing a loss function within some very high-dimensional parameter space, the convolution of both the kernels with the signals and the kernels at different layers significantly reduces the computational cost of training such a model when compared to the traditional ANN architecture. The former is known as local connectivity and the latter is known as weight sharing.

For Peer Review

The main purpose of adding a convolutional layer to a ML model is thus to exploit the low-dimensional, high-level representation of the input data. A fully connected layer can then be employed to build the mapping relationship between these high-level representations and predictive variables. In this paper, we trained a CNN to achieve a simple classification task for the fluid problem discussed in section II-B. The kernels and corresponding feature maps were then extracted to study the flow features which the model identified. A summary of the CNN architecture used in this work is shown in Fig. 4. Methods of regularization, dropout, exponential learning rate decay and moving average were used in order to avoid overfitting and improve the robustness of the model. Hyperparameters of the CNN, which came about through sensitivity studies and review of similar ML models in the literature, are summarized in Table 2.

Only information about the pressure field was provided to the neural network in this study. It was observed that the (normalized) pressure information was highly dominant in the training process even when the (normalized) velocity information was also provided to the neural network. This in itself is a significant observation, as subsonic buffet is known to stem, largely but possibly indirectly, from airfoil surface pressure gradients; the neural network likewise concluded from its observations that only the pressure fields distinguished qualitatively different types of buffet. It was clear that the neural networks were not equally valuing the velocity information for three reasons. Firstly, the flattened convolutional kernels and the corresponding feature maps did not change significantly in networks provided with the velocity and pressure information versus just the pressure information. Secondly, the kernels associated with the velocity information did not identify coherent structures in the same way that did the pressure kernels. Thirdly, a neural network provided with only velocity information did not yield dynamically significant coherent structures nearly as well as did the networks provided with only pressure information.

For Peer Review

**Table 2 CNN hyperparameters.**

| Hyperparameter | Value | Hyperparameter | Value |
|---|---|---|---|
| **Architecture of CNN** | | **Optimization of CNN** | |
| Size of square convolutional kernel | 3/5/10/20 | Optimizer | Adam |
| Number of convolutional kernel | 10 | Batch size | 100 |
| Activation function | Relu | Training steps | 10,000 |
| Stride for convolution | 1 | Learning rate base | 0.0005 |
| Stride for pooling | 2 | Learning rate decay | 0.99 |

Submitted to AIAA Journal. Confidential - Do not distribute.

| | | | |
|---|---|---|---|
| Padding for convolution | Yes | Regularization | 0.0001 |
| Padding for pooling | Yes | Moving average decay | 0.99 |
| Number of units in fully connected layer | 200 | | |
| Dropout ratio for fully connected layer | 0.5 | | |

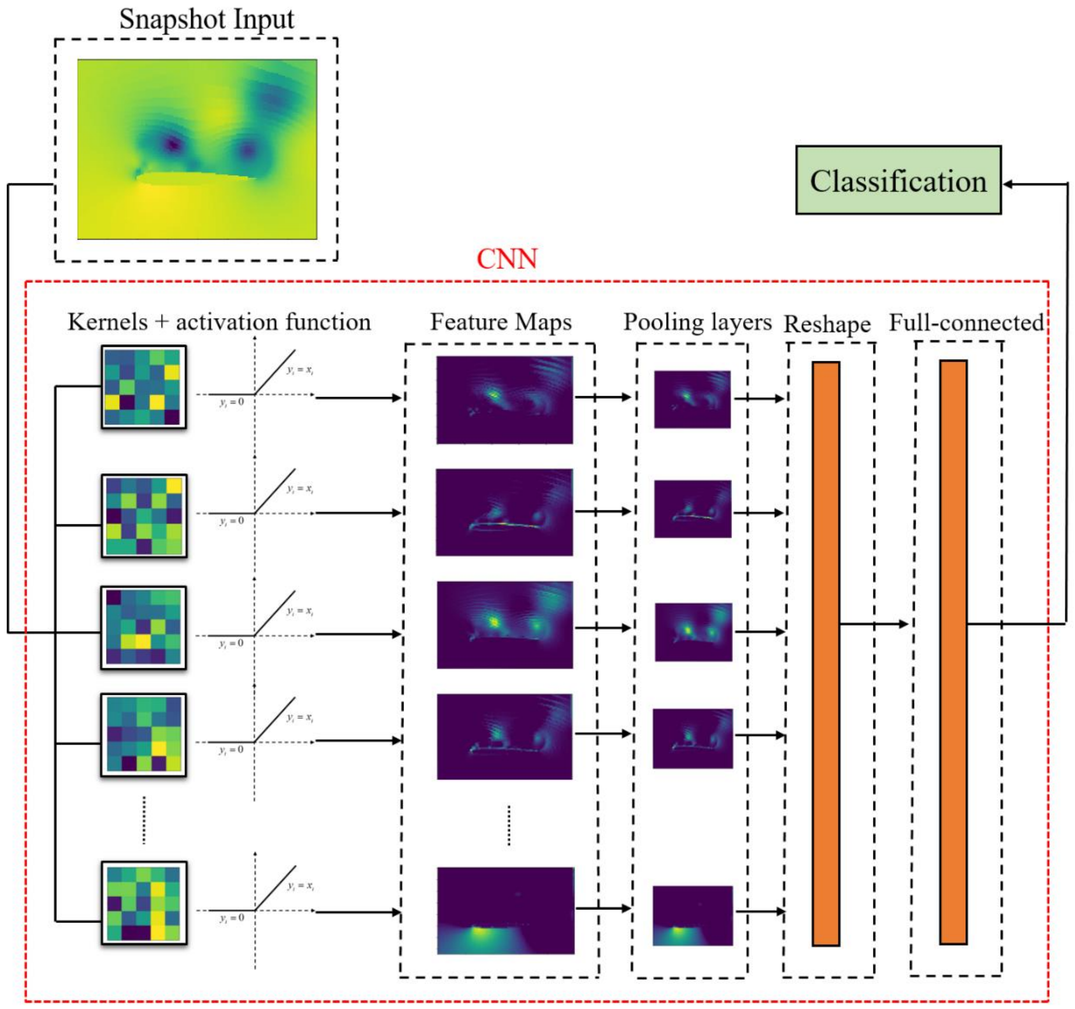


**Fig. 4 Architecture of the conventional CNN employed for buffet flow classification.**

## B. Results and Discussion

A selection of 6,000 out of the total 21,000 snapshots (detailed in Table 3) were used to train 4 CNNs, each of which had different kernel sizes as outlined in Table 2. The entire dataset (21,000 snapshots) was employed to test the trained models. The training dataset was constructed to ensure each qualitatively different flow state (see Table 1) was represented by the same number of snapshots. The training data set accounted for the different buffet states but not all

Submitted to AIAA Journal. Confidential - Do not distribute.

of the simulated Reynolds numbers. However, the trained models performed with near-perfect accuracy for the entire dataset as summarized in Fig. 5(a). Of note is that no training data was provided from the highest and most turbulent Reynolds number, whose flow was quantitatively very different from the other chaotic Reynolds numbers simulated, yet the trained model still recognized those snapshots as chaotic. This result indicates that the CNN adequately mapped the relationship between snapshots and qualitative flow states.

The few snapshots at the highest Reynolds number which were not identified correctly were observed to be qualitatively very similar to lower Reynolds number flows; as such, the coherent structures were correctly identified but the occasional appearance of simpler structures in the chaotic flow led to minor deviations from perfect classification accuracy. Further discussion of coherent structures is conducted later in this section.

**Table 3 Selection of training dataset.**

| Reynolds Number | Time region | Flow type |
|---|---|---|
| 100 | 8,000 – 10,000 | Periodic flow |
| 600 | 8,000 – 10,000 | Quasi-periodic flow |
| 10,000 | 5,000 – 7,000 | Chaotic flow |

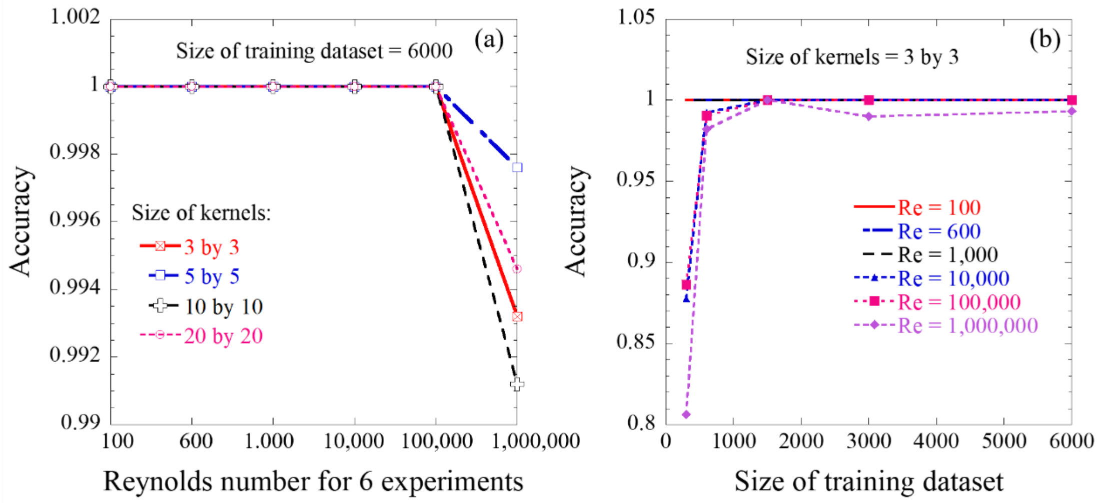


**Fig. 5 Test accuracy (a) of 6 different experiments (Reynolds numbers) for 4 CNNs with different kernel sizes and (b) of 5 different training dataset sizes for a CNN (3 by 3) with different Reynolds numbers.**

Figure 5(b) presents trends in test accuracy as a function of training dataset size at all Reynolds numbers for the smallest kernel size. The training datasets in this plot were always constructed from an equal number of snapshots from Reynolds numbers 100, 600, and 10,000; for example, for a training dataset size of 300, exactly 100 snapshots

Submitted to AIAA Journal. Confidential - Do not distribute.

were selected sequentially from each of these three Reynolds numbers. This plot indicates that the classification accuracy for periodic and quasiperiodic flows is always 100%, even with as few as 300 total snapshots being provided for training. This was expected, as the longest oscillatory period of the periodic and quasiperiodic flow states was approximately 100 consecutive snapshots. As such, no additional information is provided to the model with any more snapshots at these Reynolds numbers. The more weakly chaotic flows at Re = 10,000 and 100,000 required up to 1,200 snapshots to reach and remain at 100% classification accuracy. This indicates that there are some longer-time periodicities still dominant at these higher Reynolds numbers. Snapshots from the Re = 1,000,000 flow appeared, as expected, to yield less organized convergence behavior in terms of classification accuracy. However, the accuracy reached and remained above 0.95 when the size of training dataset was larger than 600 (200 snapshots from each type of flow). This was a remarkably small training dataset for the relatively high classification accuracy at the highest Reynolds number. This indicates that while the chaotic flow is temporally very complex, its coherent structures (as identified by the CNN) are not temporally sensitive. Such a conclusion is in agreement with the separability assumptions employed in empirical modeling methods like Galerkin POD-based reduced-order models [39].

As was discussed in section II-C, CNNs are known to be particularly successful when the input data can be decomposed into some form of hierarchical basis representation. Those low-dimensional characteristics can be extracted by convolutional kernels and visualized by the feature maps which connect the convolutional layers. Figure 6 provides an example, where a snapshot of a chaotic flow is taken as the input for the trained CNN (kernel size of 3 pixels square). The feature maps in this plot were representative of all snapshots provided to the neural network. The corresponding convolutional kernel is presented above each feature map. It is clear that each kernel identifies a certain pattern within the pressure field. Feature maps 1 and 10 were the only ones to extract the shape of the airfoil itself; these are thus denoted as "edge kernels." The 4th and 7th kernels accurately extract local, quasi-circular low- and high-pressure regions in the original snapshot, denotes as "bubble kernels." Feature maps 5, 6 and 9 extracted the high pressure region near the airfoil's leading edge, and are thus called "high pressure kernels." The remaining three feature maps (2, 3 and 8) show little coherence and are thus denoted as "useless kernels."

These results are significant as they show that the neural network automatically identified three large-scale coherent structures without human intervention or knowledge of the flow's kinematics. This is especially meaningful as only

For Peer Review

Submitted to AIAA Journal. Confidential - Do not distribute.

spatial characteristics were provided in each snapshot, without considering the temporal relationship between different snapshots for one flow regime.

Thus, the convolutional neural network assembled itself as follows. Fundamental patterns were identified at the lowest level which consistently identified the coherent structures characteristic of the qualitatively different flow regimes provided to the model. These coherent structures were then correlated to flow regime classifications in a fully connected high-level layer to a high degree of fidelity. The relative weights (also included in Fig. 6) of the feature maps indicate that different certain structures were valued more than others. The edge kernels were valued approximately as much as were the useless kernels, which means that the model considered the airfoil shape of little value for determining the type of flow regime. This aligns with thin-airfoil aerodynamic theory, from which it can be concluded that at this high incidence angle the airfoil profile minimally influences the qualitative flow characteristics. The high-pressure kernels were valued equally and with enough weight that they could influence the model’s flow classification. While one bubble kernel was a factor of five more heavily weighted than were the other kernels, the other bubble kernel was weighted comparably to the high-pressure kernels.

As such, the presence of localized fluctuations in pressure was found to most significantly inform the model’s flow classification. If the bubbles were present much more than was the high-pressure region near the airfoil’s leading edge, the flow was classified as chaotic. If the bubbles existed with comparable magnitude to the high-pressure region, the flow was quasiperiodic. If the bubbles were much less present than was the high-pressure region, then the flow was periodic. This nuanced classification algorithm, developed entirely by the neural network, aligns with an understanding of the airfoil flow’s kinematics. For example, Kurtulus [31] observed through a rigorous analysis of wake structures a similar pattern in coherent structures.

For Peer Review

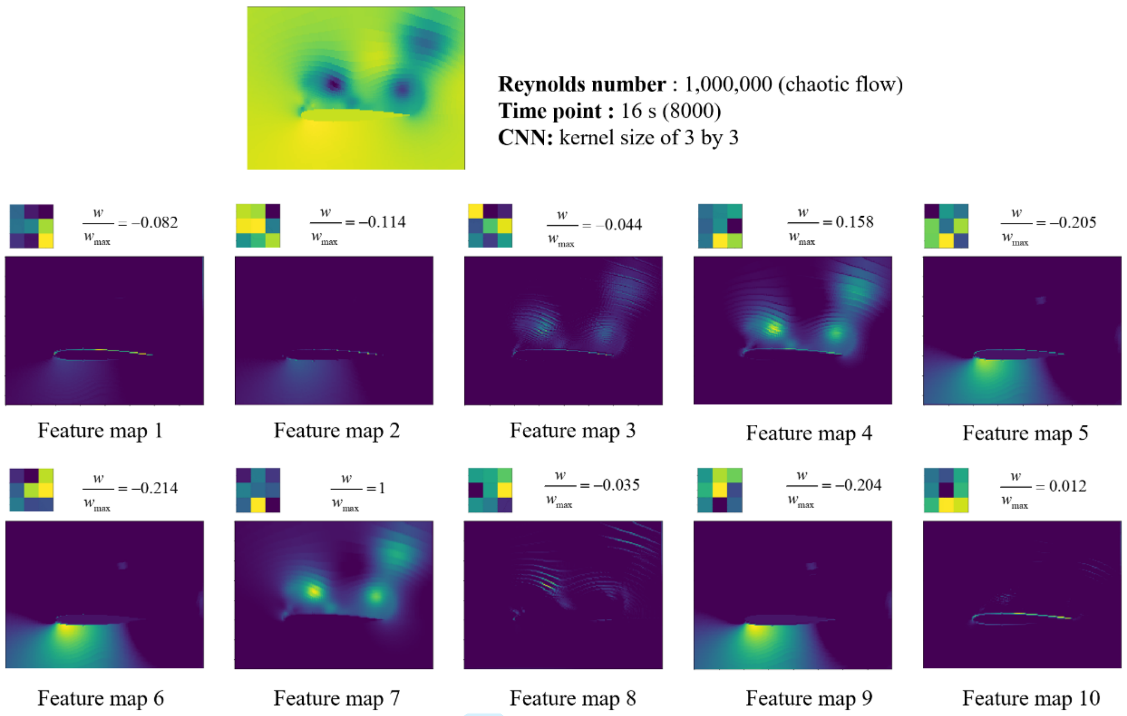


**Fig. 6 Ten kernels of the trained CNN and the corresponding feature maps for the example snapshot. The normalized weights *w* are sums of the per-pixel, per-kernel weights trained in the fully-connected layer. Animations of these feature maps are provided online.**

These results were obtained with kernels which were three pixels square; a brief study was conducted to understand the sensitivity of the coherent structure identification to kernel size. The same procedure was followed as outlined above, with only the kernel size changing. Table 4 details the number of kernels in each dynamical category, as developed above. While the edge kernels do not appear when the kernels are larger than 3 pixels square, more useless kernels appear as the kernel size increases. The dynamically valuable kernels, viz. the bubble and the high-pressure kernels, exhibited less sensitivity to kernel size but also became less common as the kernel size increased significantly. Edge detection is known to require smaller kernels, and the lack of dynamical significance of the airfoil edge does not motivate the model to try to retain the airfoil shape information. The pressure bubbles are themselves rarely larger than 20 pixels in diameter with this interpolated resolution, so again it makes sense that in a low convolutional layer the kernels would struggle to take a form which can consistently identify the bubbles. As such, the loss of coherent structures with increased kernel size was not surprising.

Although the coherent structures were not as clearly identified with the larger kernels, the large-kernel models still performed well in their classification task. This can be understood by the concept of “receptive field” in ML, which

Submitted to AIAA Journal. Confidential - Do not distribute.

eer Review

is the region of the input space that affects a particular unit of the network. In our model, we only have one convolutional layer, and the size of kernel is the value of receptive field. When increasing the value of receptive field, the information that neurons can contact is much larger, which means that the kernel can summarize more global information. Corresponding features are much more abstract than those with little kernels, whose information is organized locally and with more detail, and can therefore not be as easily understood by the users. The definition of a useless kernel is simply a kinematic one, corresponding to resulting feature maps which cannot be easily understood by a human as a known dynamically significant pattern. However, these abstract features will be very useful for a computer to distinguish the category, and that is the reason why the accuracy for them is still high.

**Table 4 Number of functional kernels for different trained CNNs.**

| Kernel size | Edge kernel | Bubble kernel | High-pressure kernel | Useless kernel |
|---|---|---|---|---|
| $3 \times 3$ | 2 | 2 | 3 | 2 |
| $5 \times 5$ | 0 | 1 | 2 | 7 |
| $10 \times 10$ | 0 | 0 | 2 | 8 |
| $20 \times 20$ | 0 | 0 | 0 | 10 |

For Peer Review

## IV. Convolutional Long-Short Term Memory Implementation

The conventional CNN identified several static coherent flow structures in several buffet flows such that it would discern with high accuracy the qualitative nature of the flow regime. These coherent structures, while spatially local by virtue of the convolutional layer, are similar to empirical modes identified using other characterization approaches like the proper orthogonal or dynamic mode decompositions. A next logical step, therefore, is to introduce temporal information into the neural model so that it can compute characteristic time scales associated with these coherent structures. It will be shown that classification accuracy is no higher in this convolutional long-short term memory (CNN-LSTM) neural network, but that further insight is gained into the dynamics of the airfoil buffet.

### A. Formulation

The LSTM network was first proposed by Sepp Hochreiter and Jurgen Schmidhuber [40] in 1997 as a variant of the recurrent neural network (RNN). A mathematical summary of the LSTM architecture is presented in the appendix.

Submitted to AIAA Journal. Confidential - Do not distribute.

It can not only process single data points (such as images), but also entire sequences (such as speech or video). The LSTM architecture improved the capability of processing long data sequences by addressing stability bottlenecks like the vanishing gradient which frustrated early RNN implementations. In section III, the input data sample for the CNN was a single temporal snapshot; only spatial characteristics were considered in the model. In this LSTM architecture, an untrained CNN was still used to convert the 2D flow snapshots into characteristic 1D vectors, and then these vectors were fed into the LSTM network chronologically. Employing a single CNN in this way differs from conventional LSTM architectures and allows kernels comparable to those in Fig. 6 to be computed from the CNN-LSTM. Because our quantifiable task is to differentiate between three different flows, only the output of the last cell is desired from the standpoint of training the model; the coherent structures identified along the way remain of fundamental interest.

The architecture of the implemented LSTM network is outlined in Fig. 7. Hyperparameters for the CNN had the same values as those in Table 2, with only kernels of size 3 by 3 being implemented for the results in this section. Other LSTM hyperparameters are outlined in Table 5.

**Table 5 LSTM hyperparameters.**

| Hyperparameter | Value |
|---|---|
| Number of units in LSTM | 200 |
| Length of sequence (snapshot) | 200 |
| Batch size | 1 |
| Dropout ratio for LSTM | 0.75 |

For Peer Review

Submitted to AIAA Journal. Confidential - Do not distribute.

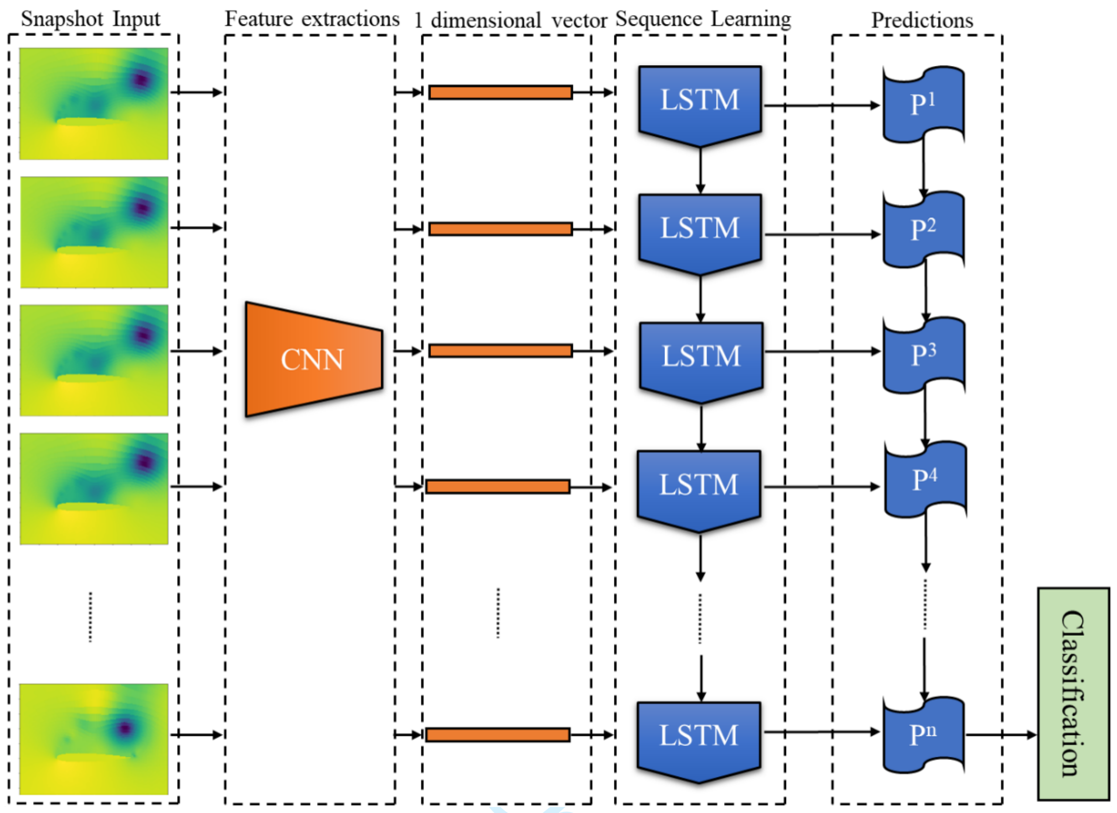


**Fig. 7 CNN-LSTM architecture overview.**

### B. Results and Discussion

The number of snapshots in one data sequence was set to 200, which means that 6,000 snapshots in the training dataset were divided into 30 data sequences. These 30 training sequences resulted in a model which was then tested on 105 test sequences comprising the entire UDNS data set outlined in section II. The resulting classification accuracy was very high, as before: the machine classified falsely 2 out of 105 data sequences after 10,000 training steps. Figure 8 shows kernels and their corresponding feature maps in the convolutional layer for the trained CNN-LSTM. Compared with those in Fig. 6, it is clear that the number of dynamically significant, viz. bubble and high-pressure, kernels increased when temporal information was considered. The edge and useless kernels, which had low ratios of weight in the CNN but still existed in the trained model, disappeared entirely in the CNN-LSTM model. Kernel 8 in Fig. 8 is the most similar to an edge kernel, but even this one contains some pressure bubble information and its weight is nonnegligible. As was the case with the CNN in section III, the bubble kernels will play a dominant role in the task of classification as their weight ratios are much larger than most of other kernels. However, the weights are now more

Submitted to AIAA Journal. Confidential - Do not distribute.

widely distributed between several subtly different bubble kernels, rather than a particular bubble kernel accomplishing a majority of the bubble-identifying task as was the case with the conventional CNN.

The CNN-LSTM model thus identified coherent structures similar to those in the conventional CNN when provided with the same training data sets, and performed comparably for the nominal classification task set forth. With the inclusion of temporal information, the CNN-LSTM model employed only kernels which captured dynamically significant characteristics; no "edge" or "useless" kernels were constructed. The trade-off, naturally, is that training a CNN-LSTM involves solving within a much larger parameter space and therefore requires significantly more computational resources to train.

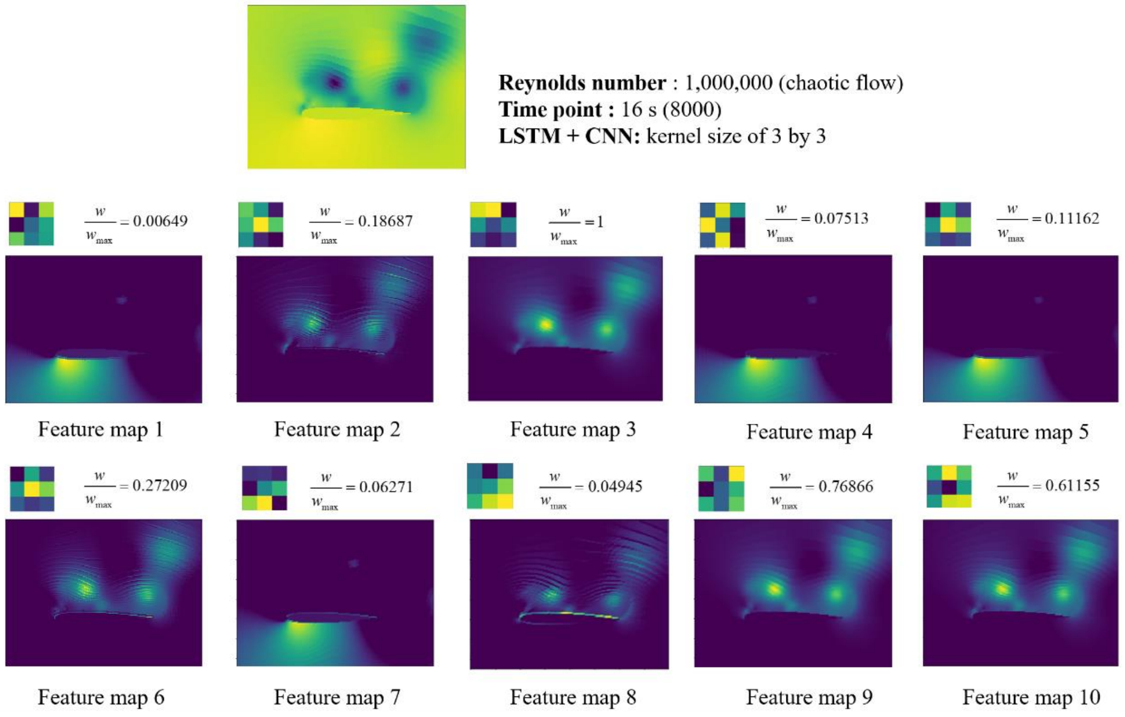


**Fig. 8 Ten kernels of the trained CNN-LSTM and the corresponding feature maps for an example snapshot. The number above every feature map is the sum of the weight for it in the fully connected layer.**

## V. Conclusions and Future Work

In this paper, a highly accurate convolutional neural network was successfully trained to recognize different manifestations of subsonic buffet over a high-incidence airfoil when provided with individual temporal snapshots. By extracting convolutional kernels and the corresponding feature maps from the trained model, the capability of identifying large-scale coherent features was validated. Sensitivity of hyperparameters, including the size of the

Submitted to AIAA Journal. Confidential - Do not distribute.

training dataset, convolutional kernel size and general network architecture, were explored. Four main conclusions are stated as follows.

1) The trained CNN automatically identified three large-scale structures, including the airfoil edge, localized shedding pressure abnormalities (viz. “bubbles”) and the high-pressure region near the airfoil’s leading edge. This was accomplished without human intervention or knowledge of the flow’s kinematics.
2) The presence of localized fluctuations in pressure, in both the bubbles and the high-pressure region, was found to inform most significantly the model’s flow classification. These were the only highly weighted characteristics in the CNN model and the only ones developed in the CNN-LSTM model.
3) Smaller convolutional kernels were necessary to identify coherent structures as understandable by human users. Larger convolutional kernels still resulted in highly accurate flow classifications, but were less physically informative to the users due to the “receptive field” concept.
4) Consideration of temporal information in the CNN-LSTM improved both classification accuracy and the dynamical insight available from the trained model. The multiscale nature of the chaotic flow was identified as dynamically important by the model, again with no provided kinematic information.

In general, it is demonstrated in this work that the CNN has the potential to extract large-scale coherent structures in complex fluid flows. In future work, other flows of engineering interest like jet flows and mixing layers will be studied and other parameters will be varied including the Mach number.

For Peer Review

## Appendix

### A. Mathematical Formulation of the CNN Architecture

In this appendix, $I(i,j)$ denotes the tensor of inputs and $K^w(m,n)$ is the tensor of the kernel $w$. In machine learning, cross-correlation operations defined in Eq. (1) are the convolutional operation central to the network architecture [42]. An example of convolving tensors in 2D without flipping is shown in Fig. 9.

$$S^w(i,j) = I * K = \sum_n \sum_m I(i+m, j+n) \;\; K^w(m,n) \tag{1}$$

The corresponding feature map $F^w$ is defined as

$$F^w = \varphi(S^w + b^w) \tag{2}$$

Submitted to AIAA Journal. Confidential - Do not distribute.

where $\varphi(z) = \mathrm{Relu}(z) = \max(0, z)$ denotes a rectified linear unit – “relu” – activation function, which is used to introduce a nonlinear relationship into the model. Also, $b^w$ is the bias for each kernel $w$. In the pooling layer, the process can be defined as

$$P^w_{u,v} = \max\left(F^w_{h,j}\right) \quad | \quad u \le h \ \ \& \ \ j \le v + 2 \tag{3}$$

where $h, j$ is the corresponding cover range of the pooling kernel in feature map $F^w$. An example of the pooling process is also shown in Fig. 9.

After the pooling layer, the map is then reshaped into a one-dimensional vector $p_{f1}$. Then, a weight matrix is introduced with the form

$$W = \begin{bmatrix} w_{11} & \cdots & w_{1f} \\ \vdots & \ddots & \vdots \\ w_{r1} & \cdots & w_{rf} \end{bmatrix} \tag{4}$$

where $r$ represents the number of kernels in this layer. The layer is known as fully connected because all elements of the pooled vector may interact through the weight matrix. The output of the fully connected layer is then defined as

$$p_{f2} = \varphi\left(W \cdot \ p_{f1} + b'\right) \tag{5}$$

$$\Phi = W' \cdot p_{f2} + b'' \tag{6}$$

where $\varphi$ is the same activation function as before and $b'$ and $b''$ are the biases in the fully connected layer. Note that there is also an additional weight matrix $W'$. The vector $\Phi$ is the output of neural network for forward propagation.

For Peer Review

Submitted to AIAA Journal. Confidential - Do not distribute.

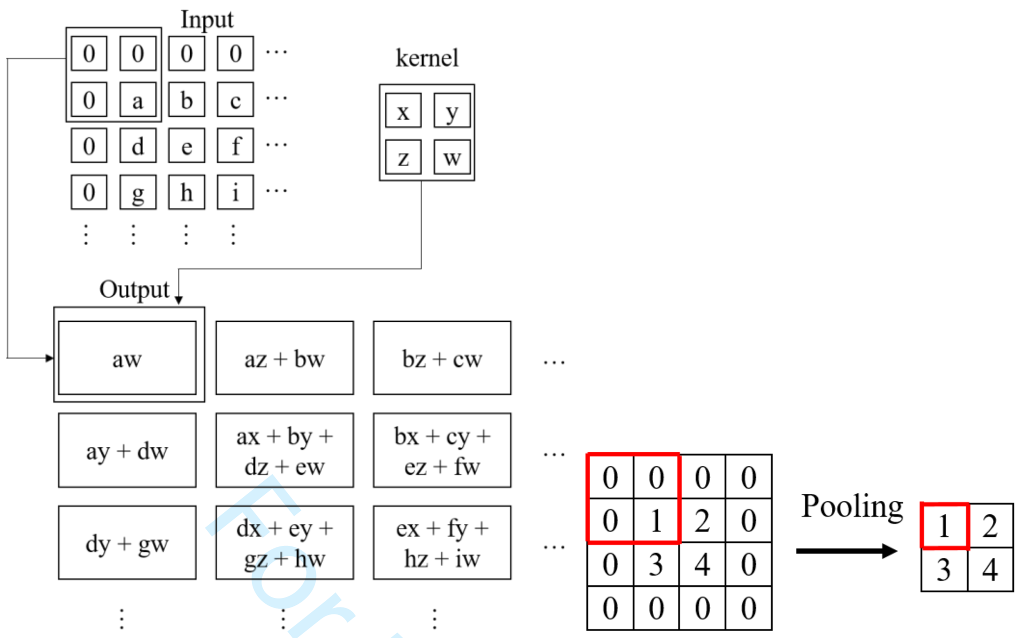


**Fig. 9 Convolution in two dimensions without kernel flip (left) and the pooling process for feature maps with a stride of 2 (right).**

For Peer Review

## B. Architecture of a LSTM Cell

The typical LSTM cell contains three gates at a step $t$: an input gate $i_t$, an output gate $o_t$ and a forget gate $f_t$. The structure of an LSTM cell is illustrated in Fig. 9, and corresponding equations for computing these gates are as follows [43, 44]:

$$
\begin{aligned}
f_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
\widetilde{C_t} &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
C_t &= f_t * C_{t-1} + i_t * \widetilde{C_t} \\
h_t &= o_i * \tanh(C_t)
\end{aligned}
\tag{7}
$$

Where $C_t$ and $\widetilde{C_t}$ denote the cell state and the updated cell state, respectively. These states will propagate ahead through the network. The cell input at a given step is denoted as $x_t$ and output at that step $t$ is given by $h_t$. The matrices $W$ are the weights in each gate. The sigmoid function $\sigma$ and the tanh function are both activation functions which introduce nonlinear in a manner similar to the relu function did previously. The flow of training information in a LSTM network is regulated through these gates by adding information (input gate), removing (forget gate) or passing

Submitted to AIAA Journal. Confidential - Do not distribute.

it to the next cell (output gate). Detailed information can be found in Ref [40] with a valuable summary of the organization in Ref [43]. Figure 10, adapted from Ref [43], outlines the overall structure of the LSTM architecture.

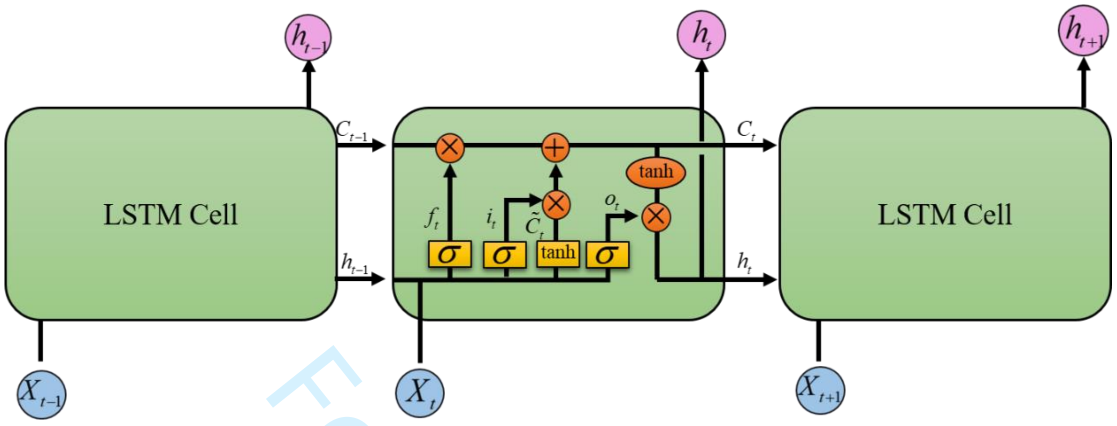


**Fig. 10 Architecture of a LSTM Cell, adapted from [43].**

For Peer Review

## Funding Sources

This research is supported by the Duke Aeroelasticity Group and is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1644868.

## Acknowledgments

The Duke computing cluster and its staff are acknowledged for enabling the high-fidelity flow simulations to be performed. Appreciation is extended to Dr. Lawrence Carin for his insightful online instructional resources regarding machine learning and neural networks.

## References

[1] Y. Zhuang, H.-j. Tan, Y.-z. Liu, Y.-c. Zhang, Y. Ling, High resolution visualization of Görtler-like vortices in supersonic compression ramp flow, Journal of Visualization, 20 (2017) 505-508.

[2] Y. Zhuang, H. Tan, W. Wang, X. Li, Y. Guo, Fractal features of turbulent/non-turbulent interface in a shock wave/turbulent boundary-layer interaction flow, Journal of Fluid Mechanics, 869 (2019).

Submitted to AIAA Journal. Confidential - Do not distribute.

[3] M. Mustafa, N. Parziale, M. Smith, E. Marineau, Amplification and structure of streamwise-velocity fluctuations in compression-corner shock-wave/turbulent boundary-layer interactions, Journal of Fluid Mechanics, 863 (2019) 1091-1122.

[4] S. Brunton, B. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, arXiv preprint arXiv:1905.11075, (2019).

[5] K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence modeling in the age of data, Annual review of fluid mechanics, 51 (2019) 357-377.

[6] B.D. Tracey, K. Duraisamy, J.J. Alonso, A machine learning strategy to assist turbulence model development, in: 53rd AIAA aerospace sciences meeting, 2015, pp. 1287.

[7] Z.J. Zhang, K. Duraisamy, Machine learning methods for data-driven turbulence modeling, in: 22nd AIAA Computational Fluid Dynamics Conference, 2015, pp. 2460.

[8] A.P. Singh, S. Medida, K. Duraisamy, Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils, AIAA Journal, (2017) 2215-2227.

[9] A.P. Singh, K. Duraisamy, Using field inversion to quantify functional errors in turbulence closures, Physics of Fluids, 28 (2016) 045110.

[10] E.J. Parish, K. Duraisamy, A paradigm for data-driven predictive modeling using field inversion and machine learning, Journal Of Computational Physics, 305 (2016) 758-774.

[11] K. Duraisamy, Z.J. Zhang, A.P. Singh, New approaches in turbulence and transition modeling using data-driven techniques, in: 53rd AIAA Aerospace Sciences Meeting, 2015, pp. 1284.

[12] J.-X. Wang, J.-L. Wu, H. Xiao, Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data, Physical Review Fluids, 2 (2017) 034603.

[13] J.-L. Wu, J.-X. Wang, H. Xiao, J. Ling, A priori assessment of prediction confidence for data-driven turbulence modeling, Flow, Turbulence and Combustion, 99 (2017) 25-46.

[14] J. Ling, A. Kurzawski, J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, Journal of Fluid Mechanics, 807 (2016) 155-166.

[15] L. Zhu, W. Zhang, J. Kou, Y. Liu, Machine learning methods for turbulence modeling in subsonic flows around airfoils, Physics of Fluids, 31 (2019) 015105.

For Peer Review

Submitted to AIAA Journal. Confidential - Do not distribute.

[16] J. Schmidhuber, Deep learning in neural networks: An overview, Neural networks, 61 (2015) 85-117.

[17] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 2625-2634.

[18] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-c. Woo, Convolutional LSTM network: A machine learning approach for precipitation nowcasting, in: Advances in neural information processing systems, 2015, pp. 802-810.

[19] Y. Zhang, W. Sung, D. Mavris, Application of convolutional neural network to predict airfoil lift coefficient, in: AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2018, American Institute of Aeronautics and Astronautics Inc, AIAA, 2018.

[20] X. Guo, W. Li, F. Iorio, Convolutional neural networks for steady flow approximation, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 481-490.

[21] S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, S. Kaushik, Prediction of aerodynamic flow fields using convolutional neural networks, Computational Mechanics, (2019) 1-21.

[22] A.T. Mohan, D.V. Gaitonde, A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks, arXiv preprint arXiv:1804.09269, (2018).

[23] D. Tang, E.H. Dowell, Experimental aerodynamic response for an oscillating airfoil in buffeting flow, AIAA Journal, 52 (2014) 1170-1179.

[24] F.M. Besem, J.D. Kamrass, J.P. Thomas, D. Tang, R.E. Kielb, Vortex-induced vibration and frequency lock-in of an airfoil at high angles of attack, Journal of Fluids Engineering, 138 (2016) 011204.

[25] T. Zhou, S.-S. Feng, E.H. Dowell, Buffeting and Lock in of an Airfoil at High Angle of Attack, Journal of Aircraft, 55 (2017) 771-780.

[26] J.W. Jaworski, E.H. Dowell, Scaling Analysis for Aeroelastic Phenomena Using the Navier-Stokes Fluid Model, AIAA Journal, 50 (2012) 2622-2626.

[27] C. Rumsey, B. Smith, G. Huang, Description of a website resource for turbulence modeling verification and validation, in: 40th Fluid Dynamics Conference and Exhibit, 2010, pp. 4742.

For Peer Review

Submitted to AIAA Journal. Confidential - Do not distribute.

[28] Bastos, Kai M. Kruger, Dowell, Earl H. (2019). The buffet phenomenon and limit cycle oscillations. The International Forum on Aeroelasticity and Structural Dynamics.

[29] R. Wiebe, L. Virgin, A heuristic method for identifying chaos from frequency content, Chaos: An Interdisciplinary Journal of Nonlinear Science, 22 (2012) 013136.

[30] E. Rossi, A. Colagrossi, G. Oger, D. Le Touzé, Multiple bifurcations of the flow over stalled airfoils when changing the Reynolds number, Journal of Fluid Mechanics, 846 (2018) 356-391.

[31] D.F. Kurtulus, On the unsteady behavior of the flow around NACA 0012 airfoil with steady external conditions at Re= 1000, International Journal of Micro Air Vehicles, 7 (2015) 301-326.

[32] J.L. McClelland, D.E. Rumelhart, P.R. Group, Parallel distributed processing, MIT press Cambridge, MA:, 1987.

[33] H.B. Demuth, M.H. Beale, O. De Jess, M.T. Hagan, Neural network design, Martin Hagan, 2014.

[34] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE, 86 (1998) 2278-2324.

[35] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556, (2014).

[36] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097-1105.

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1-9.

[38] A.D. Beck, D.G. Flad, C.-D. Munz, Deep neural networks for data-driven turbulence models, arXiv preprint arXiv:1806.04482, (2018).

[39] K. Taira, S.L. Brunton, S.T.M. Dawson, C.W. Rowley, T. Colonius, B.J. McKeon, O.T. Schmidt, S. Gordeyev, V. Theofilis, L.S. Ukeiley, Modal Analysis of Fluid Flows: An Overview, AIAA Journal, 55 (2017) 4013-4041.

[40] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation, 9 (1997) 1735-1780.

[41] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike

For Peer Review

Submitted to AIAA Journal. Confidential - Do not distribute.

Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[42] D. Papp, Prediction of unsteady nonlinear aerodynamic loads using deep convolutional neural networks: Investigating the dynamic response of agile combat aircraft, (2018).

[43] C.Olah, "Understanding LSTM networks." http://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015.

[44] K. Li, J. Kou, W. Zhang, Deep neural network for unsteady aerodynamic and aeroelastic modeling across multiple Mach numbers, Nonlinear Dynamics, 96 (2019) 2157-2177.

For Peer Review

Submitted to AIAA Journal. Confidential - Do not distribute.