

## Highlights

### **Convolutional Neural Networks for Feature Identification in Complex Fluid Flows**

Shizheng Wen, Michael W. Lee, Kai M. Kruger Bastos, Earl H. Dowell

- Convolutional neural networks are used for pattern recognition to a subsonic airfoil experiencing buffet at a high incidence angle and a range of Reynolds numbers.
- Gradient-weighted class activation mapping is used to identify the importance of coherent structures constructed as part of the neural network.
- The learned flow patterns align with the existing theories on the dynamics of subsonic buffet flows.
- The network can give a more discrete task to identify the flow regime from individual spatial flow snapshots.

# Convolutional Neural Networks for Feature Identification in Complex Fluid Flows

Shizheng Wen<sup>a,\*</sup>, Michael W. Lee<sup>b</sup>, Kai M. Kruger Bastos<sup>b</sup> and Earl H. Dowell<sup>b</sup>

<sup>a</sup>Nanjing University of Aeronautics and Astronautics, Yudao Street 29, Nanjing, 210016, China

<sup>b</sup>Duke University, Durham, 27708, North Carolina, USA

## ARTICLE INFO

### Keywords:

subsonic buffet flows  
feature identification  
convolutional neural network  
long-short term memory

## ABSTRACT

Recent efforts have shown machine learning to be useful for the prediction 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 confounding dynamics. In this paper, convolutional neural networks (CNNs) were trained to recognize several qualitatively different subsonic buffet flows over a high-incidence airfoil, and a near-perfect accuracy was performed with only a small training dataset. 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. An approach named Gradient-weighted Class Activation Mapping (Grad-CAM) was then applied to the trained model to indicate the importance of these feature maps in classification. Sensitivity to hyperparameters including network architecture and convolutional kernel size was also explored, and results show that smaller kernels are better at coherent structure identification than are larger kernels. A long-short term memory CNN was subsequently used to demonstrate that with the inclusion of temporal information, the coherent structures remained qualitatively comparable to those of the conventional CNN. 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. Introduction


In recent years, with the development of high-performance computing architectures and experimental measurement capabilities Zhuang, Tan, Liu, Zhang and Ling (2017); Zhuang, Tan, Wang, Li and Guo (2019); Mustafa, Parziale, Smith and Marineau (2019), fluid researchers are able to obtain high precision and high-resolution spatiotemporal data of large-scale fluid 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 Brunton, Noack and Koumoutsakos (2019); Duraisamy, Iaccarino and Xiao (2018).

For nonlinear fluid flow regimes, incorporating domain knowledge into learning algorithms has been demonstrated to be feasible. Data-driven turbulence modeling presents promising extensions to more conventional numerical system closure techniques and are therefore of significant value for engineering applications Duraisamy et al. (2018). For example, Tracey et al. Tracey, Duraisamy and Alonso (2015) successfully reproduced the Spalart-Allmaras model by replacing the deliberately removed source term with machine-learned functional forms. Duraisamy's group also pursued efforts in data-driven turbulence modeling Zhang and Duraisamy (2015); Singh, Medida and Duraisamy (2017); Singh and Duraisamy (2016); Parish and Duraisamy (2016);

Duraisamy, Zhang and Singh (2015); Duraisamy (2021) 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 Wang, Wu and Xiao (2017); Wu, Wang, Xiao and Ling (2017). Ling et al. Ling, Kurzawski and Templeton (2016) 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. Zhu, Zhang, Kou and Liu (2019) 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 Schmidhuber (2015); Donahue, Anne Hendricks, Guadarrama, Rohrbach, Venugopalan, Saenko and Darrell (2015); Shi, Chen, Wang, Yeung, Wong and Woo (2015) like the convolutional neural network (CNN) and the long-short term memory network (LSTM) provide tools for researchers to evaluate the temporal and spatial patterns in data. For example, Ye et al. Ye, Zhang, Song, Wang, Chen and Huang (2020) applied CNN to predict the pressure coefficient on a cylinder from velocity distributions in its wake flow. Zhang's group explored many potential applications of deep learning modeling for unsteady aerodynamics and aeroelasticity Kou and Zhang (2021). Zhang et al. Zhang, Sung and Mavris (2018) trained

\*Corresponding author

 wenshizheng0710@gmail.com (S. Wen)

ORCID(s): 0000-0000-0000-0000 (S. Wen)

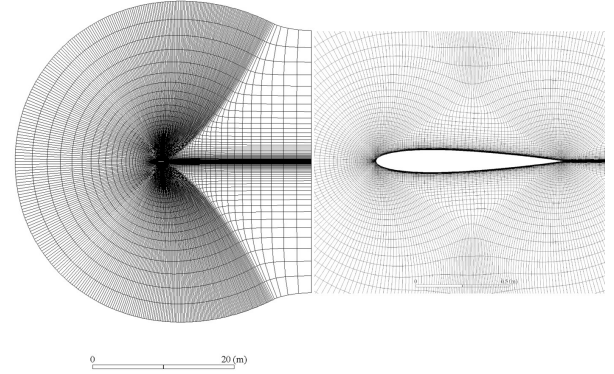
multiple linked CNNs to learn the lift coefficients of an airfoil with a variety of shapes in multiple flow regimes. Guo et al. Guo, Li and Iorio (2016) proposed a convolutional encoder-decoder approach that can predict steady velocity and pressure fields. Bhatnagar et al. Bhatnagar, Afshar, Pan, Duraisamy and Kaushik (2019) improved the computational efficiency of this effort by sharing encoder-decoder layers. Mohan et al. Mohan and Gaitonde (2018) built a deep learning approach to reduced order modeling (ROM) for isotropic turbulent flows by replacing Galerkin projection with LSTM neural networks.

The success of the aforementioned works indicates the encouraging prospects of ML in the 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 Ye et al. (2020); Colvert, Alsaman and Kanso (2018). 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 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: 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 with only small training dataset, 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.

## 2. Problem Formulation

Flow over a NACA 0012 airfoil was computed via direct numerical simulation (DNS) 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.



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

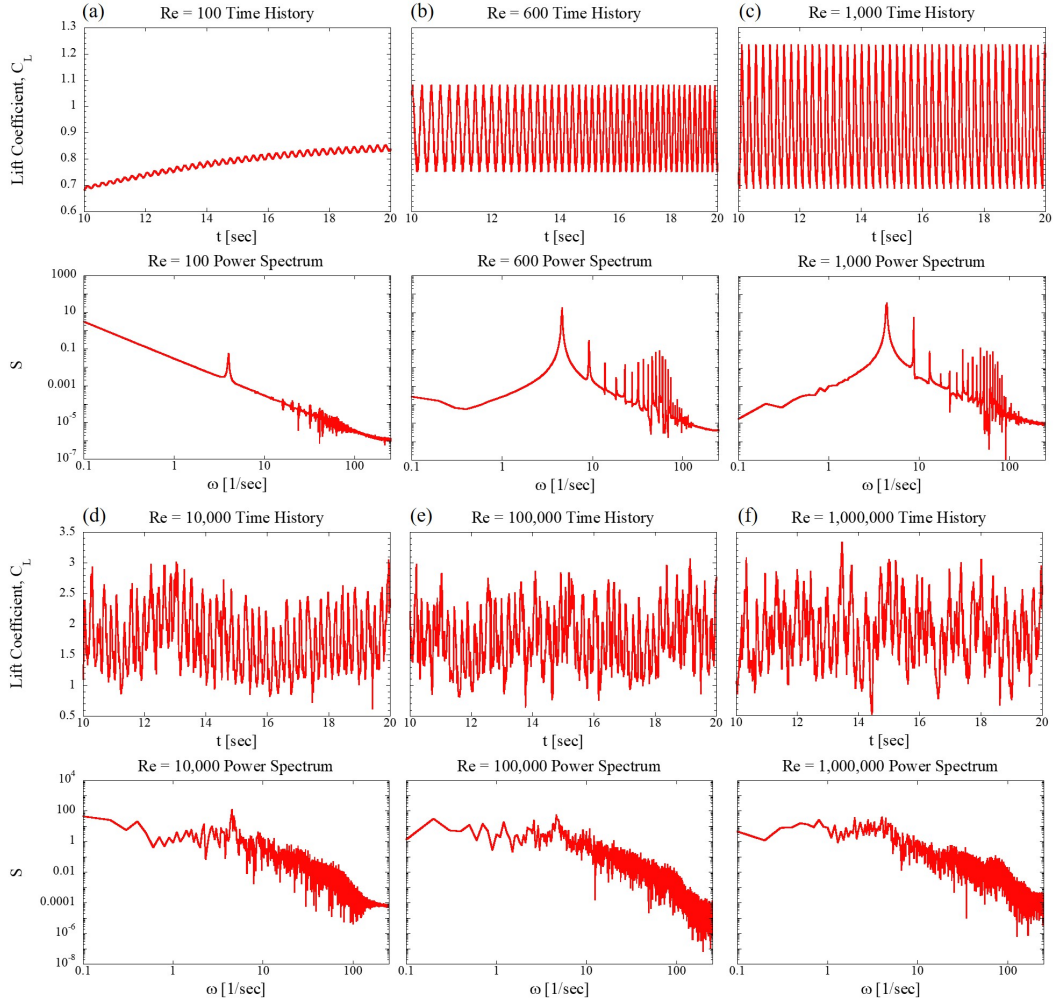
### 2.1. Subsonic Buffet

The buffet phenomenon in subsonic open flows has received recent attention Tang and Dowell (2014); Besem, Kamrass, Thomas, Tang and Kielb (2016). Vortex shedding similar to bluff body flows is observed in high-incidence flows around stationary airfoils Tang and Dowell (2014); Besem et al. (2016); Zhou, Feng and Dowell (2018). Lift coefficients are one way to characterize these time-periodic dynamic instabilities. A scaling analysis of the Navier-Stokes equations Jaworski and Dowell (2012) demonstrates that both the peak-to-peak oscillating lift and reduced frequency in buffeting flows are of order unity. Consequently, 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.

### 2.2. DNS 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 air foil'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 31 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 Bastos (2020).



**Figure 2:** Response of lift coefficient changing with time and corresponding power spectrum at Reynolds numbers of 100, 600, 1,000, 10,000, 100,000 and 1,000,000.

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. Simulations were

performed with a truncated NASA grid Rumsey, Smith and Huang (2010), 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 Kühnen, Song, Scarselli, Budanur, Riedl, Willis, Avila and Hof (2018). 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 27 experimental study, the lift and drag coefficient reduced-frequencies were observed to be in good agreement between the DNS simulation and experimental results.

Figure 2 shows steady-state time histories and power spectra of the lift coefficient ( $C_L$ ) as computed at different Reynolds numbers. By employing the heuristic flow characterization developed by Wiebe and Virgin Wiebe and Virgin (2012) and in agreement with existing literature Rossi, Colagrossi, Oger and Le Touzé (2018), 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. Consequently, 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 Bastos (2020) determined that there was weak dependence on grid resolution for large ranges of Reynolds number when seeking the lift and drag coefficient reduced frequencies. Consequently, 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.

**Table 1**

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

Reynolds Number	Sampling region	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

### 2.3. 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 DNS 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 also 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 Kurtulus (2015) were used to determine what points were within the airfoil area.

## 3. Convolutional Neural Network Implementation

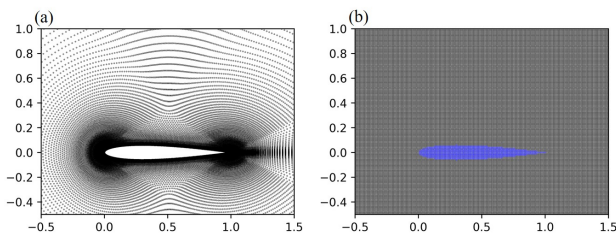
A convolutional neural network was employed to identify the large-scale coherent structures associated with the

qualitatively different manifestations of buffet. All neural networks in this paper were constructed within the Google tensorflow environment. A summary of the mathematics of the CNN are presented in the appendix A. With only small training dataset, 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. Grad-CAM was subsequently used to indicate the importance of these identified flow features in the presence of chaotic flow.

### 3.1. Formulation

The artificial neural networks (ANNs) first became prominent in the field of artificial intelligence in the 1980s McClelland, Rumelhart, Group et al. (1987). They are straightforward abstractions of biological neurons, realized as elements in an artificial network like a program or silicon-based circuits Demuth, Beale, De Jess and Hagan (2014). 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) Gu, Wang, Kuen, Ma, Shahroudy, Shuai, Liu, Wang, Wang, Cai et al. (2018). LeCun et al. LeCun, Bottou, Bengio and Haffner (1998) 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 Simonyan and Zisserman (2014); Krizhevsky, Sutskever and Hinton (2012); Szegedy, Liu, Jia, Sermanet, Reed, Anguelov, Erhan, Vanhoucke and Rabinovich (2015).

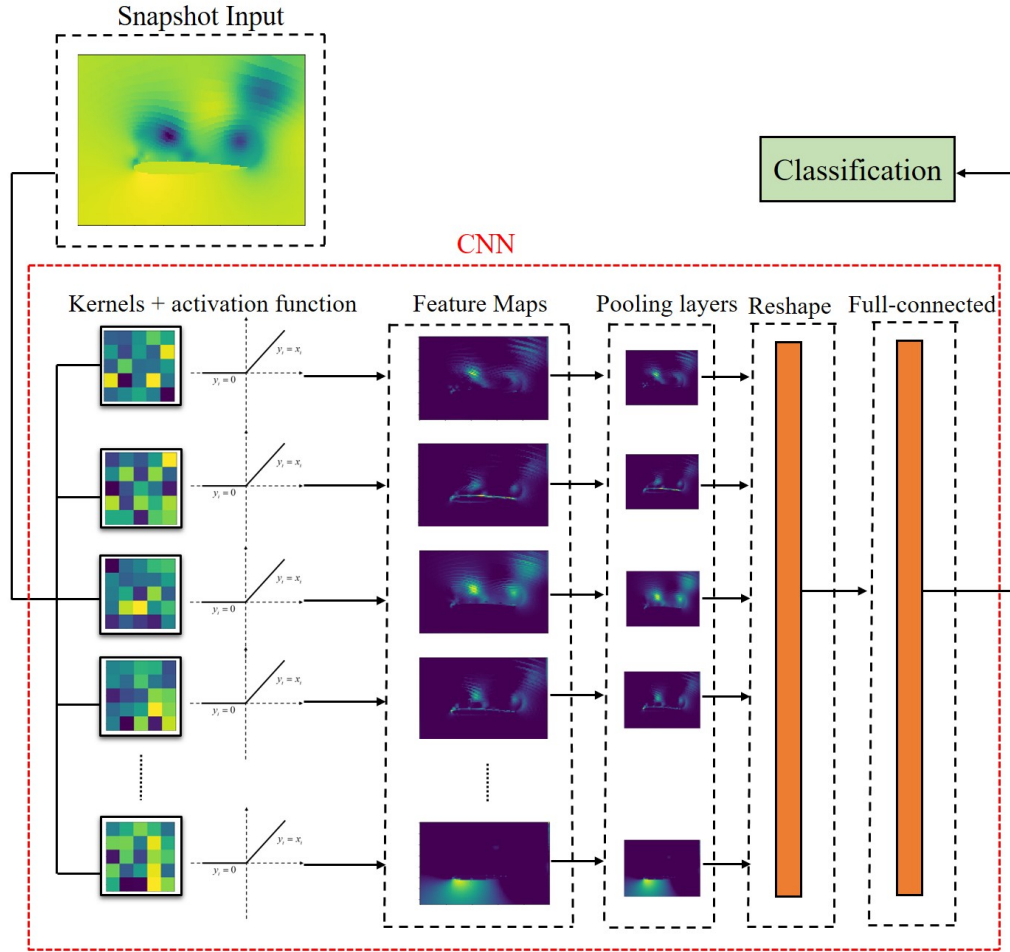
CNNs derive their utility from seeking patterns at multiple scales through the process of convolving small kernels – perhaps five pixels square – with a much larger input signal – perhaps a grayscale image of 100 pixels square. CNNs thus are particularly successful when the input data can be decomposed into some form of hierarchical basis representation; and this is often labeled as automatic feature extraction Beck, Flad and Munz (2019). The optimization problem stems from finding kernels which identify the most fundamental patterns in the signal, which can then combine



**Figure 3:** DNS grid subsection (a) and the Cartesian grid (b) provided to the neural networks.

**Table 2**  
CNN hyperparameters.

Hyperparameter	Value	Hyperparameter	Value
<b>Architecture of CNN</b>		<b>Optimization of CNN</b>	
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
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		



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

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.

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.

### 3.2. Results and Discussion

A selection of 600 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 remaining dataset (20,400 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 was very small and accounted for the different buffet states but not all of the simulated Reynolds numbers. However, the trained models performed with high accuracy (over 0.95) for the test dataset. This was a remarkably small training dataset for the relatively high classification accuracy especially at the highest Reynolds number. 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. What's more, though 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 Taira, Brunton, Dawson, Rowley, Colonius, McKeon, Schmidt, Gordeyev, Theofilis and Ukeiley (2017).

**Table 3**

Selection of training dataset.

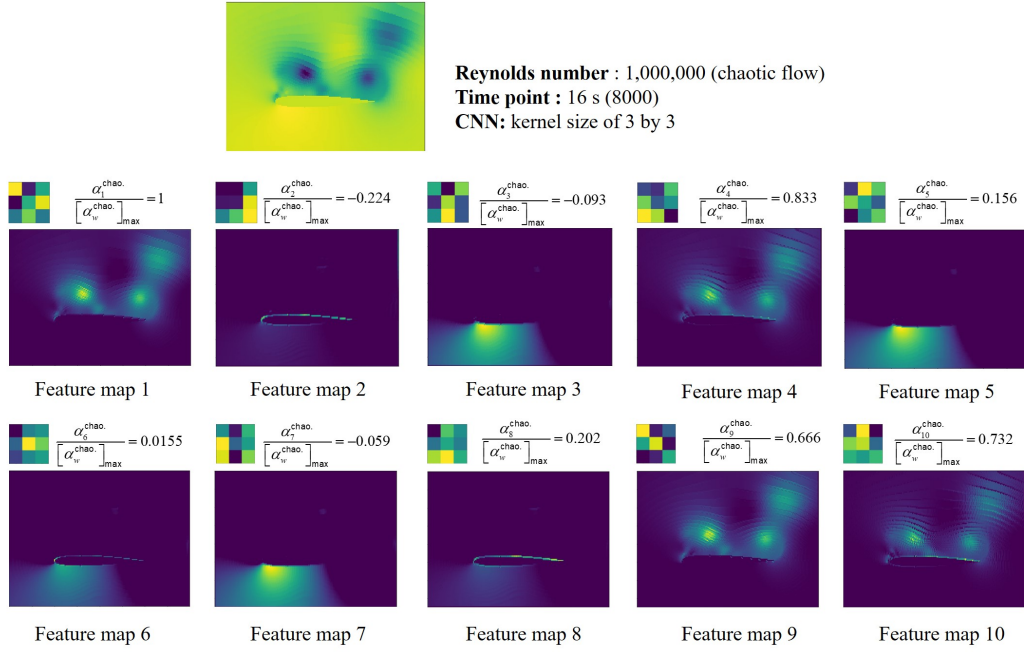
Reynolds Number	Sampling region	Flow type
100	8000-10,000	Periodic flow
600	8000-10,000	Quasi-periodic flow
10,000	5000-10,000	Chaotic flow

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; Consequently, 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.

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 5 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 2 and 8 were the only ones to extract the shape of the airfoil itself; these are thus denoted as "edge kernels." The 1th, 4th, 9th and 10th kernels accurately extract local, quasi-circular low- and high-pressure regions in the original snapshot, denotes as "bubble kernels." Feature maps 3, 5 and 7 extracted the high pressure region near the airfoil's leading edge, and are thus called "high pressure 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 spatial characteristics were provided in each snapshot, without considering the temporal relationship between different snapshots for one flow regime.

The convolutional neural network thus 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



**Figure 5:** Ten kernels of the trained CNN and the corresponding feature maps for the example snapshot. The normalized neuron importance weights denotes  $\alpha_w^{\text{chao.}} / \left[ \alpha_w^{\text{chao.}} \right]_{\max}$ , where the maximum value  $\left[ \alpha_w^{\text{chao.}} \right]_{\max}$  is  $\alpha_1^{\text{chao.}}$ . Animations of these feature maps are provided online.

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. In order to fully understand the importance of these identified patterns for classification, a technique called Gradient-weighted Class Activation Mapping (Grad-CAM) Selvaraju, Cogswell, Das, Vedantam, Parikh and Batra (2017) was applied to the trained model for obtaining the neuron importance weights  $\alpha_w^c$ , which captures the importance of feature map  $w$  for a target class  $c$ :

$$\alpha_w^c = \frac{1}{Z} \sum_i \sum_j \underbrace{\frac{\partial y^c}{\partial F_{ij}^w}}_{\text{gradient via backprop}} \quad (1)$$

The number of target class  $c$  in our model is three (periodic flow  $c$ : perio., quasi-periodic flow  $c$ : qua. and chaotic flow  $c$ : chao.).  $\frac{\partial y^c}{\partial F_{ij}^w}$  represents the gradient of the score for class  $c$ ,  $y^c$ , with respect to a pixel  $F_{ij}$  in feature maps  $w$  of a convolutional layer.  $Z$  represents the number of pixels in the feature map. Of note is that only feature map with positive values will be emphasized, larger values in weights means higher importance of these features in classification. Here, we calculated weights of ten feature maps for chaotic flow. The relative weights (also included in Fig. 5) of the feature maps indicate that different certain structures were valued more than others for the emergence of chaotic flow. The edge kernels were less valued for the occurrence of

buffet flow, 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 also unvalued for the right classification of buffet flow. All bubble kernels were more heavily weighted than were the other kernels.

Consequently, 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 Kurtulus (2015) observed through a rigorous analysis of wake structures a similar pattern in coherent structures.

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 5 pixels square, more useless



**Table 4**

Number of functional kernels for different trained CNNs.

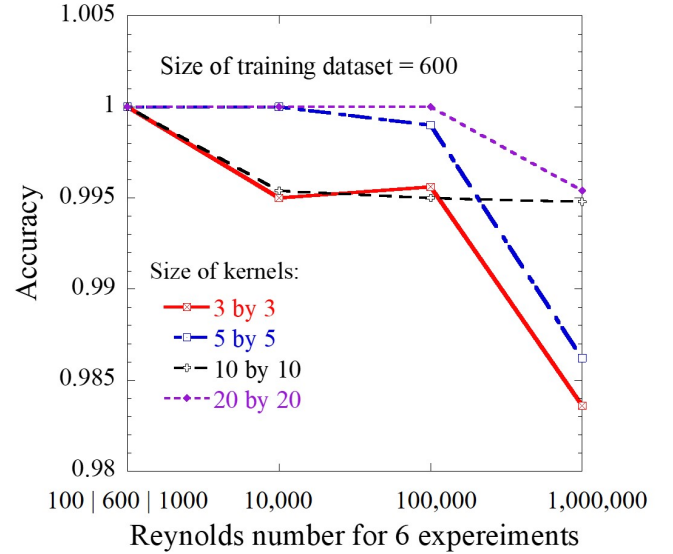
Kernel size	Edge Kernel	Bubble kernel	High-pressure kernel	Useless kernel
$3 \times 3$	2	4	4	0
$5 \times 5$	1	5	2	2
$10 \times 10$	0	1	2	7
$20 \times 20$	0	0	0	10

kernels, showing little coherence and are thus denoted as “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. Consequently, 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 as summarized in Fig. 6. This can be understood by the concept of “receptive field” in ML, which 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.

#### 4. 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. In the appendix B, proper orthogonal modes were also calculated. A next logical step, therefore, is to introduce



**Figure 6:** Test accuracy of 6 different experiments (Reynolds numbers) for 4 CNNs with different kernel sizes.

temporal information into the neural model so that it can compute characteristic time scales associated with these coherent structures.

##### 4.1. Formulation

The LSTM network was first proposed by Sepp Hochreiter and Jurgen Schmidhuber (1997) in 1997 as a variant of the recurrent neural network (RNN). A mathematical summary of the LSTM architecture is presented in the appendix C. 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. 5 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;

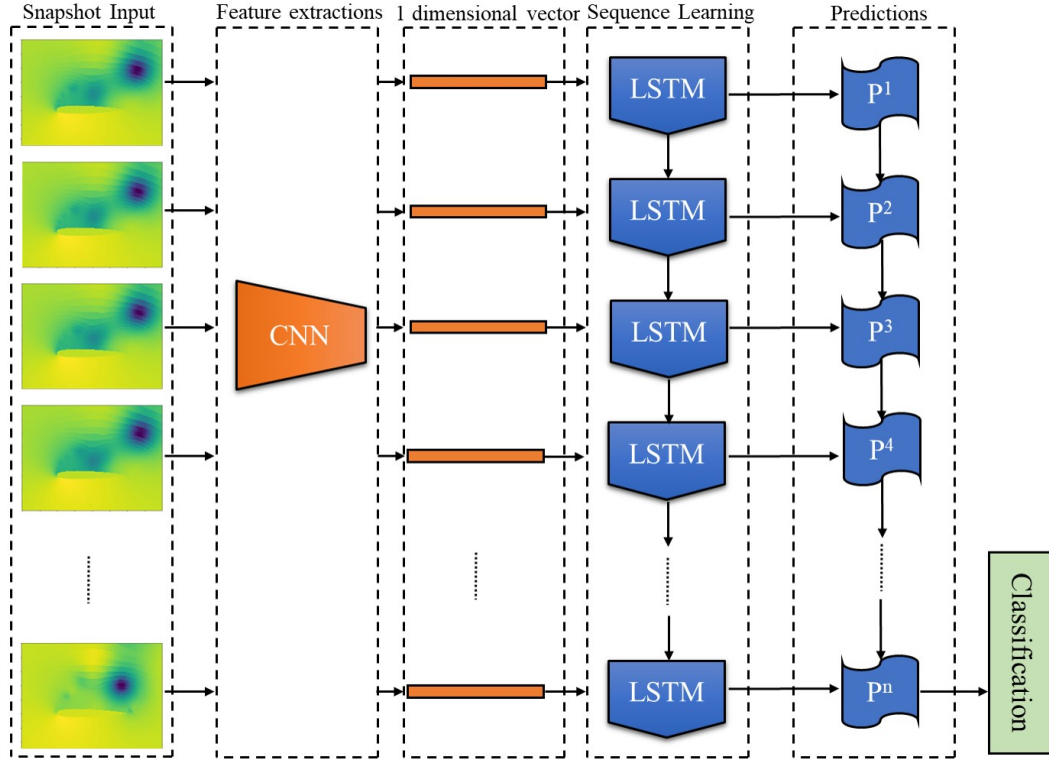


Figure 7: CNN-LSTM architecture overview.

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)	20
Batch size	1
Dropout ratio for LSTM	0.75

#### 4.2. Results and Discussion

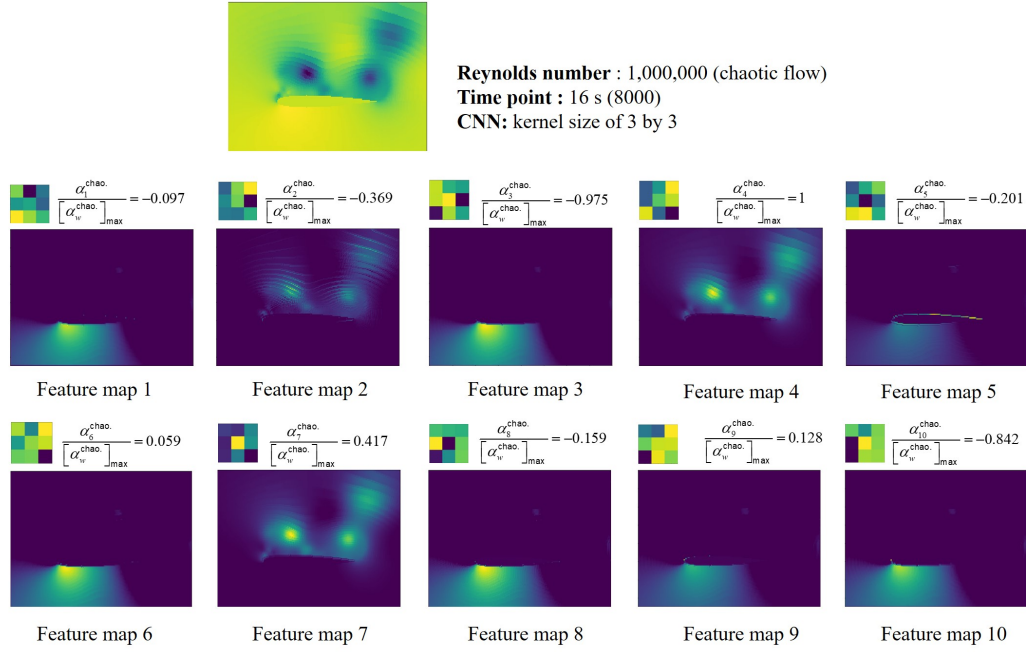
The number of snapshots in one data sequence was set to 20, which means that 600 snapshots in the training dataset were divided into 30 data sequences. These 30 training sequences resulted in a model which was then tested on 1050 test sequences comprising the entire DNS data set outlined in section II. The resulting classification accuracy was nearly perfect: the machine classified falsely 1 out of 1050 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. 5, it is clear that the CNN also successfully identified three large-scale coherent structures. As was the case with the CNN in section III, the bubble kernels (kernel 4 and 7) will play a dominant role in the task of classification as their neuron importance weights are much larger than most of other kernels. Of note is that although the kernel 2 also have the tendency to identify the bubbles, its importance in classification is negligent as the result of its less dynamical information.

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 have nearly perfect accuracy for test dataset. 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.

#### 5. Conclusions and Future Works

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 training dataset,



**Figure 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 normalized neuron importance weights.

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 (bubbles) was found to inform most significantly the model’s flow classification. These were both highly weighted characteristics in the CNN model and 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 the classification accuracy. 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 to achieve specific tasks in complex fluid flows. By applying Grad-CAM technique to the trained model, we can know the relative importance of these identified structures in the task we concerned, which can help us gain further insights into these confounding dynamics from the perspective of

machine. This cannot be realized from existing modal decomposition techniques (e.g. POD, DMD). In future work, we believed these “feature-filter” kernels can be preserved for transfer learning to advance our training process in other complex fluid flows. Keeping the parameters in the convolutional layers unchanged while shifting parameters in the full-connected layers can be a promising path. And other flows of engineering interest like jet flows and mixing layers will be studied and other parameters will be varied including the Mach number.

## 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. (2) are the convolutional operation central to the network architecture Gu et al. (2018). An example of convolving tensors in 2D without flipping is shown in Fig. 9.

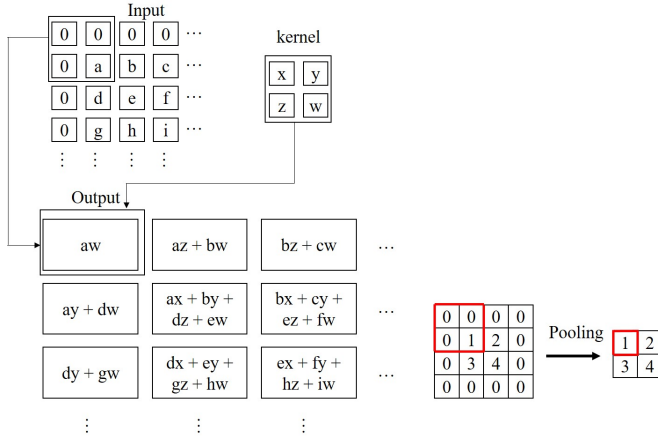
$$S^w(i, j) = I \cdot K = \sum_n \sum_m I(i + m, j + n) K^w(m, n) \quad (2)$$

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

$$F^w = \varphi(S^w + b^w) \quad (3)$$

where  $\varphi(z) = \text{Relu}(z) = \max(0, z)$  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

$$P_{u,v}^w = \max(F_{h,j}^w) \mid u \leq h \text{ \& } j \leq v + 2 \quad (4)$$



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

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

$$W = \begin{bmatrix} w_{11} & \cdots & w_{1f} \\ \vdots & \ddots & \vdots \\ w_{r1} & \cdots & w_{rf} \end{bmatrix} \quad (5)$$

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

$$p_{f2} = \varphi(W \cdot p_{f1} + b') \quad \Phi = W' \cdot p_{f2} + b'' \quad (6)$$

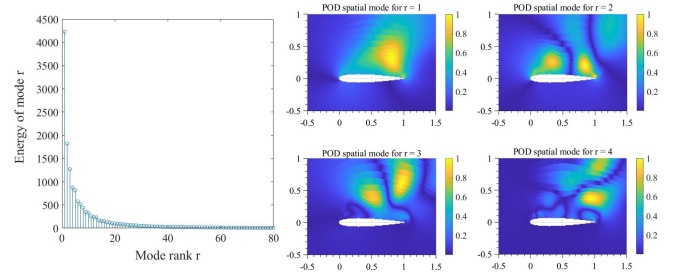
where  $\varphi$  is the same same activation function as before and  $b'$  and  $b''$  are the biases in the fully connected layer. Note that there is also an additional weight matrix  $W'$ . The vector  $\Phi$  is the output of neural network for forward propagation.

## B. Results of Spatial Modes from Proper Orthogonal Decomposition (POD)

Proper orthogonal decomposition is a method of a data matrix decomposition and approximation. The concept of it on a discrete data set (a matrix) coincides with the singular value decomposition (SVD). The original matrix  $D$  can be written as a product of three matrices:

$$D = U \Sigma V^T \quad (7)$$

where the matrices  $U$  and  $V$  are both orthonormal, and matrix  $\Sigma$  is diagonal with elements on the diagonal denoted as  $\sigma_i$ . In POD, matrices  $U$ ,  $\Sigma$  and  $V^T$  have a special physical meaning. A product of corresponding columns of the three component matrices is called a mode. Matrix  $U$  is a matrix of



**Figure 10:** Energy of every decomposed mode and POD spatial modes for  $r = 1, 2, 3, 4$ .

spatial structures, which give a “shape” to the mode. Matrix  $\Sigma$  is a matrix of amplitudes, which specifies the importance of every mode in the solution. Matrix  $V^T$  is a matrix of temporal structures and gives dynamics to the mode. The goal of POD is to reduce the rank of data matrix and find a series of mode matrixes to approximate it.

Once the decomposition is made, the original matrix of rank  $d$  can be approximated by a matrix  $\widetilde{D}_r$  of a lower rank  $r$ , which can be written as a sum of “simple” matrices  $A_i$  multiplied by the corresponding amplitude  $\sigma_i$ , extracted from the diagonal of matrix  $\Sigma$ .

$$D \approx \widetilde{D}_r = A_1 \sigma_1 + A_2 \sigma_2 + \cdots + A_r \sigma_r \quad (8)$$

In our question, we select snapshots of time region from 8,000 to 8,200 at Reynolds number 1,000,000 to construct our original matrix  $D$ . Energy of every decomposed mode and POD spatial modes from 1 to 4 was also calculated, as illustrated in Fig. 10. These spatial modes are very similar to the coherent structures identified by the “bubble kernels” in the CNN.

## C. 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 Christopher (2015); Li, Kou and Zhang (2019):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (9a)$$

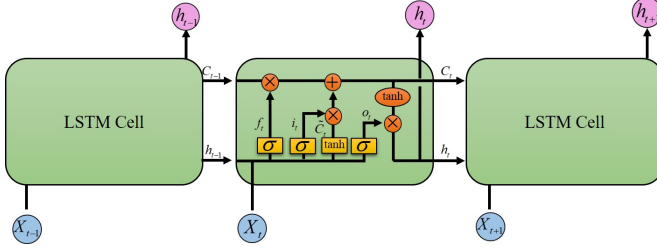
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (9b)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (9c)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (9d)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (9e)$$





**Figure 11:** Architecture of a LSTM Cell, adapted from Ref. Li et al. (2019)

$$h_t = o_t \cdot \tanh(C_t) \quad (9f)$$

Where  $C_t$  and  $\tilde{C}_t$  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 it to the next cell (output gate). Detailed information can be found in Ref Hochreiter and Schmidhuber (1997) with a valuable summary of the organization in Ref Christopher (2015). Figure 10, adapted from Ref Christopher (2015), outlines the overall structure of the LSTM architecture.

## References

Bastos, K.M.K., 2020. Computational Studies of Buffet and Fluid-Structure Interaction in Various Flow Regimes. Ph.D. thesis. Duke University.

Beck, A., Flad, D., Munz, C.D., 2019. Deep neural networks for data-driven les closure models. *Journal of Computational Physics* 398, 108910.

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

Bhatnagar, S., Afshar, Y., Pan, S., Duraisamy, K., Kaushik, S., 2019. Prediction of aerodynamic flow fields using convolutional neural networks. *Computational Mechanics* 64, 525–545.

Brunton, S., Noack, B., Koumoutsakos, P., 2019. Machine learning for fluid mechanics. *arXiv preprint arXiv:1905.11075*.

Christopher, O., 2015. Understanding lstm networks. *Understanding LSTM Networks-Colah's Blog*. Github.

Colvert, B., Alsaman, M., Kansa, E., 2018. Classifying vortex wakes using neural networks. *Bioinspiration & biomimetics* 13, 025003.

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

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

Duraisamy, K., 2021. Perspectives on machine learning-augmented reynolds-averaged and large eddy simulation models of turbulence. *Physical Review Fluids* 6, 050504.

Duraisamy, K., Iaccarino, G., Xiao, H., 2018. Turbulence modeling in the age of data. *arXiv preprint arXiv:1804.00183*.

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

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al., 2018. Recent advances in convolutional neural networks. *Pattern Recognition* 77, 354–377.

Guo, X., Li, W., Iorio, F., 2016. Convolutional neural networks for steady flow approximation, in: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 481–490.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural computation* 9, 1735–1780.

Jaworski, J.W., Dowell, E.H., 2012. Scaling analysis for aeroelastic phenomena using the navier-stokes fluid model. *AIAA journal* 50, 2622–2626.

Kou, J., Zhang, W., 2021. Data-driven modeling for unsteady aerodynamics and aeroelasticity. *Progress in Aerospace Sciences* 125, 100725.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25.

Kühnen, J., Song, B., Scarselli, D., Budanur, N.B., Riedl, M., Willis, A.P., Avila, M., Hof, B., 2018. Destabilizing turbulence in pipe flow. *Nature Physics* 14, 386–390.

Kurtulus, D.F., 2015. 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, 301–326.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 2278–2324.

Li, K., Kou, J., Zhang, W., 2019. Deep neural network for unsteady aerodynamic and aeroelastic modeling across multiple mach numbers. *Nonlinear Dynamics* 96, 2157–2177.

Ling, J., Kurawski, A., Templeton, J., 2016. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics* 807, 155–166.

McClelland, J.L., Rumelhart, D.E., Group, P.R., et al., 1987. *Parallel Distributed Processing, Volume 2: Explorations in the Microstructure of Cognition: Psychological and Biological Models*. volume 2. MIT press.

Mohan, A.T., Gaitonde, D.V., 2018. A deep learning based approach to reduced order modeling for turbulent flow control using lstm neural networks. *arXiv preprint arXiv:1804.09269*.

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

Parish, E.J., Duraisamy, K., 2016. A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of Computational Physics* 305, 758–774.

Rossi, E., Colagrossi, A., Oger, G., Le Touzé, D., 2018. Multiple bifurcations of the flow over stalled airfoils when changing the reynolds number. *Journal of Fluid Mechanics* 846, 356–391.

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

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. *Neural networks* 61, 85–117.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization, in: *Proceedings of the IEEE international conference on computer vision*, pp. 618–626.

Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c., 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems* 28.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, A.P., Duraisamy, K., 2016. Using field inversion to quantify functional errors in turbulence closures. *Physics of Fluids* 28, 045110.

- Singh, A.P., Medida, S., Duraisamy, K., 2017. Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. *AIAA journal* 55, 2215–2227.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.
- Taira, K., Brunton, S.L., Dawson, S.T., Rowley, C.W., Colonius, T., McKeon, B.J., Schmidt, O.T., Gordeyev, S., Theofilis, V., Ukeiley, L.S., 2017. Modal analysis of fluid flows: An overview. *Aiaa Journal* 55, 4013–4041.
- Tang, D., Dowell, E.H., 2014. Experimental aerodynamic response for an oscillating airfoil in buffeting flow. *AIAA journal* 52, 1170–1179.
- Tracey, B.D., Duraisamy, K., Alonso, J.J., 2015. A machine learning strategy to assist turbulence model development, in: *53rd AIAA aerospace sciences meeting*, p. 1287.
- Wang, J.X., Wu, J.L., Xiao, H., 2017. Physics-informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on dns data. *Physical Review Fluids* 2, 034603.
- Wiebe, R., Virgin, L., 2012. A heuristic method for identifying chaos from frequency content. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 22, 013136.
- Wu, J.L., Wang, J.X., Xiao, H., Ling, J., 2017. A priori assessment of prediction confidence for data-driven turbulence modeling. *Flow, Turbulence and Combustion* 99, 25–46.
- Ye, S., Zhang, Z., Song, X., Wang, Y., Chen, Y., Huang, C., 2020. A flow feature detection method for modeling pressure distribution around a cylinder in non-uniform flows by using a convolutional neural network. *Scientific reports* 10, 1–10.
- Zhang, Y., Sung, W.J., Mavris, D.N., 2018. Application of convolutional neural network to predict airfoil lift coefficient, in: *2018 AIAA/ASCE/AHS/ASC structures, structural dynamics, and materials conference*, p. 1903.
- Zhang, Z.J., Duraisamy, K., 2015. Machine learning methods for data-driven turbulence modeling, in: *22nd AIAA Computational Fluid Dynamics Conference*, p. 2460.
- Zhou, T., Feng, S.S., Dowell, E.H., 2018. Buffeting and lock in of an airfoil at high angle of attack. *Journal of Aircraft* 55, 771–780.
- Zhu, L., Zhang, W., Kou, J., Liu, Y., 2019. Machine learning methods for turbulence modeling in subsonic flows around airfoils. *Physics of Fluids* 31, 015105.
- Zhuang, Y., Tan, H., Wang, W., Li, X., Guo, Y., 2019. Fractal features of turbulent/non-turbulent interface in a shock wave/turbulent boundary-layer interaction flow. *Journal of Fluid Mechanics* 869.
- Zhuang, Y., Tan, H.j., Liu, Y.z., Zhang, Y.c., Ling, Y., 2017. High resolution visualization of görtler-like vortices in supersonic compression ramp flow. *Journal of Visualization* 20, 505–508.