Theoretical and Applied Mechanics Letters 13 (2023) 100482

Contents lists available at ScienceDirect

Theoretical and Applied Mechanics Letters

journal homepage: www.elsevier.com/locate/taml

Full Article

Feature identification in complex fluid flows by convolutional neural networks

Shizheng Wen[a,*], Michael W. Lee[b], Kai M. Kruger Bastos[c], Ian K. Eldridge-Allegra[d], Earl H. Dowell[d]

[a] Department of Mathematics, ETH Zurich, Zürich 8092, Switzerland
[b] NASA Langley Research Center, Hampton, VA 23666, USA
[c] Rivian, San Francisco, CA 94080, USA
[d] Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708, USA

ARTICLE INFO

Keywords:
Subsonic buffet flows
Feature identification
Convolutional neural network
Long-short term memory

ABSTRACT

Recent advancements have established machine learning's utility in predicting nonlinear fluid dynamics, with predictive accuracy being a central motivation for employing neural networks. However, the pattern recognition central to the networks function is equally valuable for enhancing our dynamical insight into the complex fluid dynamics. In this paper, a single-layer convolutional neural network (CNN) was trained to recognize three qualitatively different subsonic buffet flows (periodic, quasi-periodic and chaotic) over a high-incidence airfoil, and a near-perfect accuracy was obtained 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. Sensitivity to hyperparameters including network architecture and convolutional kernel size was also explored. 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, 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 [1–3]. Turbulence modeling and, more generally, nonlinear fluid dynamics has been one proving ground for neural networks [4–8].

For nonlinear fluid flow regimes, incorporating domain knowledge [9,10] 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 [4,11–13]. For example, Tracey et al. [14] successfully reproduced the Spalart-Allmaras model by replacing the deliberately removed source term with machine-learned functional forms. Duraisamy et al. also pursued efforts in data-driven turbulence modeling [15] 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. Wang et al. emphasized the physical constraint of Reynolds stress and proposed the concept of a physics-informed machine learning approach [16,17]. Zhu et al. [18] 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 [19–21] 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. [22] applied CNN to predict the pressure coefficient on a cylinder from velocity distributions in its wake flow. Kou et al. explored many potential applications of deep learning modeling for unsteady aerodynamics and aeroelasticity [5]. Zhang et al. [23] trained multiple linked CNNs to learn the lift coefficients of an airfoil with a variety of shapes in multiple flow regimes. Guo et al. [24] proposed a convolutional encoder-decoder approach that can predict steady velocity and pressure fields. Mohan et al. [25] built a deep learning approach

* Corresponding author.
E-mail address: shiwen@ethz.ch (S. Wen).

https://doi.org/10.1016/j.taml.2023.100482
Received 29 September 2023; Received in revised form 6 November 2023; Accepted 13 November 2023
Available online 14 November 2023

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 [22,26]. 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 2.1, the problem of subsonic buffet and the computation of the flow data provided to the neural networks are discussed. In Sections 2.2 and 2.3, neural network architectures including CNN and LSTM are summarized. In Section 3, comparative results and the discussion of CNN and LSTM are included. Finally, we summarize our main result and future work in Section 4.

2. Problem formulation and methodology

2.1. Problem formulation

2.1.1. Subsonic buffet

The buffet phenomenon in subsonic open flows, characterized by large-scale, time-periodic dynamic instabilities, has received increasing attention in recent research [27,28]. These instabilities, often observed in high-incidence flows around stationary airfoils, exhibit vortex shedding patterns similar to those found in bluff body flows [27–29], which plays a significant role in influencing airfoil performance. Despite its critical impact on aerodynamic performance, the dynamics of subsonic buffet flow are not fully understood, necessitating further exploration and analysis. This phenomenon exhibits nonlinear behaviors, which makes the flow difficult to predict and intractable with analytical methods. Making these predictions early in the design process for airfoils, wings, and full system is computational costly, even when relying on a linearized stability analysis. Recent research provides the prevailing theory of buffet as a global instability [30]. This global nature makes prediction of buffet difficult without detailed flow simulations. Consequently, while the dynamics 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.

2.1.2. Under-resolved direct numerical 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 airfoils lift coefficient reached a limit cycle oscillation with a positive mean value. The method of simulating such a flow is an active area of research [31,32]. Unsteady reynolds-averaged navier-stokes (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 under-resolved 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 [32].

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	8000–10000	Periodic flow
600	8000–10000	Quasi-periodic flow
1000	8000–10000	Quasi-periodic flow
10,000	5000–10000	Chaotic flow
100,000	5000–10000	Chaotic flow
1,000,000	5000–10000	Chaotic flow

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; it instead relies on the implicit averaging from the mesh to estimate the small-scale effects. 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, 1000, 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 s 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 [33], with 257 airfoil surface points. The grid extends 20 chord lengths in all directions, which was determined to be adequate to resolve far-field behavior [34]. View of the grid and specific settings used for the simulation are shown in the supplementary material. 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 Dowells experimental study [27], the lift and drag coefficient reduced-frequencies were observed to be in good agreement between the DNS simulation and experimental results. By employing the heuristic flow characterization developed by Wiebe and Virgin [35] and in agreement with existing literature [36], each Reynolds number is associated with qualitatively different flow regimes: periodic flow, quasi-periodic flow and chaotic flow, as shown in Table 1. Steady-state time histories and power spectra of the lift coefficient (C_L) as computed at different Reynolds numbers are shown in the supplementary material. All simulations were performed with a time step of 0.002 s. Consequently, the 2000 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 [31] 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. This is consistent with the spectral separation between buffet and small-scale turbulent behavior – the former's frequency scales with the airfoil chord, while the latter has frequencies that scale with the boundary-layer thickness. It is these local but large-scale flow features that this study has identified by using ML.

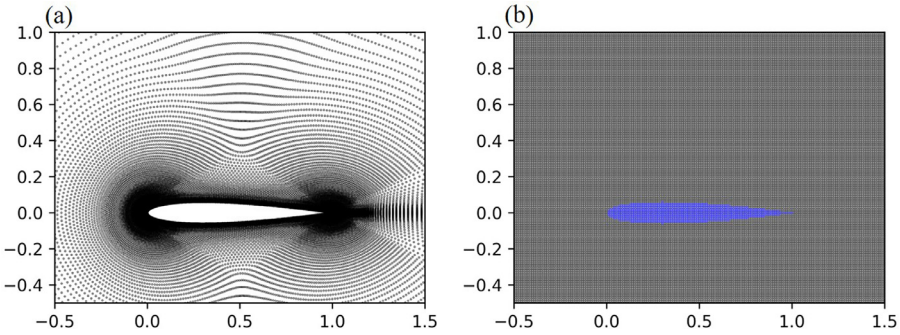

Fig. 1. DNS grid subsection (a) and the Cartesian grid (b) provided to the neural networks.

2.1.3. 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. 1a. 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. 1b. Flow values within the airfoil (the blue area in Fig. 1b) area were set to zero. Splines defined by the NACA airfoil standard [37] were used to determine what points were within the airfoil area.

2.2. Convolutional neural networks

For the classification of different flow regimes, a convolutional neural network (CNN) was implemented. This CNN was designed to extract features from temporal snapshots of the flow, enabling the identification of large-scale coherent structures inherent to the buffet phenomenon. The model construction and training are conducted using the PyTorch framework, and a detailed mathematical description of CNN is provided in the supplementary material.

In the context of this study, the CNN was structured to minimize the cross-entropy loss function for a three-class classification problem, represented as

$$L(\theta) = -\sum_{i=1}^{N}\sum_{c=1}^{M} y_{ic}\lg\left(p_{ic}(\theta)\right), \tag{1}$$

where $L(\theta)$ denotes the loss function over the parameters θ, N is the number of samples, M is the number of classes ($M = 3$ for our case), y_{ic} is the binary indicator of whether class label, c is the correct classification for observation i, and $p_{ic}(\theta)$ is the predicted probability of observation i being of class c, given the parameters θ. The architecture of our CNN is illustrated in Fig. 2. 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 2.1. The kernels and corresponding feature maps were then extracted to study the flow features which the model identified. 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.

In order to fully understand the importance of these identified patterns for classification, a technique called Gradient-weighted Class Activation Mapping (Grad-CAM) [38] was applied to the trained model for obtaining the neuron importance weights α_w^c, which captures the importance of feature map w for a target class c:

$$\alpha_w^c = \overbrace{\frac{1}{Z}\sum_i\sum_j \underbrace{\frac{\partial y^c}{\partial F_{ij}^w}}_{\text{gradient via backprop}}}^{\text{global average pooling}}. \tag{2}$$

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.

Notably, the CNN was exclusively provided with normalized pressure field data, based on the observation that pressure information was more influential in the training process and classification performance. This aligns with the understanding that subsonic buffet is predominantly influenced by surface pressure gradients. The networks's discernment that pressure fields alone sufficed to distinguish buffet types reinforces this point.

2.3. Convolutional long-short term memory implementation

The LSTM network was first proposed by Sepp Hochreiter and Jurgen Schmidhuber [39] in 1997 as a variant of the recurrent neural network (RNN). A mathematical summary of the LSTM architecture is presented in the supplementary material. 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 2.2, 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 computed from the CNN. 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. 3. 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 model. Other LSTM hyperparameters are outlined in Table 3.

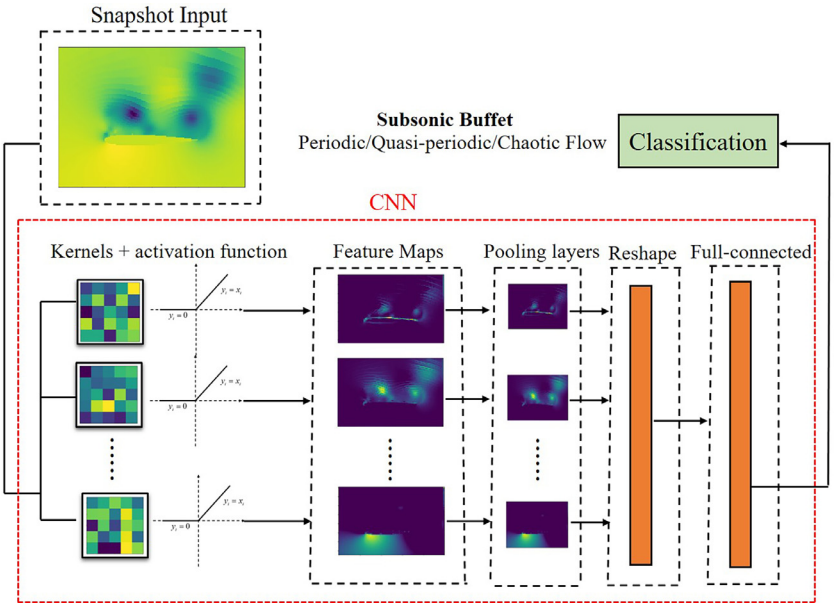

Fig. 2. Architecture of the conventional CNN employed for buffet flow classification.

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	10000
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		

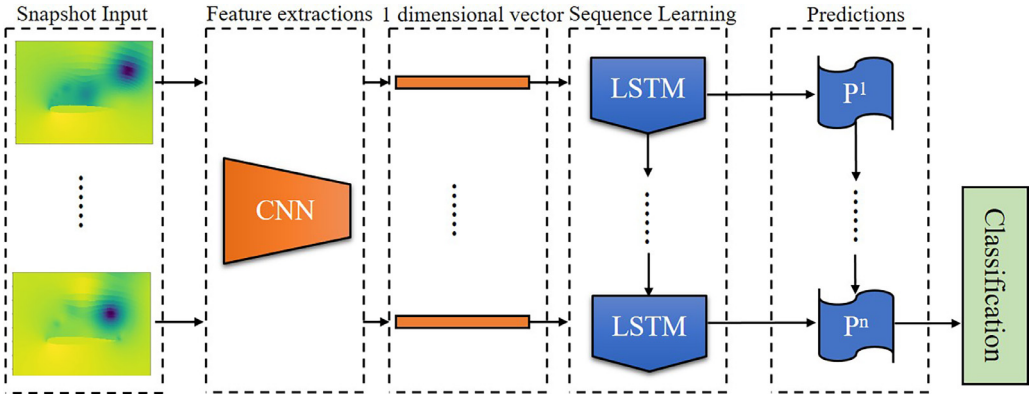

Fig. 3. CNN-LSTM architecture overview.

3. Results and discussion

3.1. Convolutional neural networks

A selection of 600 out of the total 21000 snapshots (detailed in Table 4) were used to train 4 CNNs, each of which had different kernel sizes as outlined in Table 2. The remaining dataset (20400 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

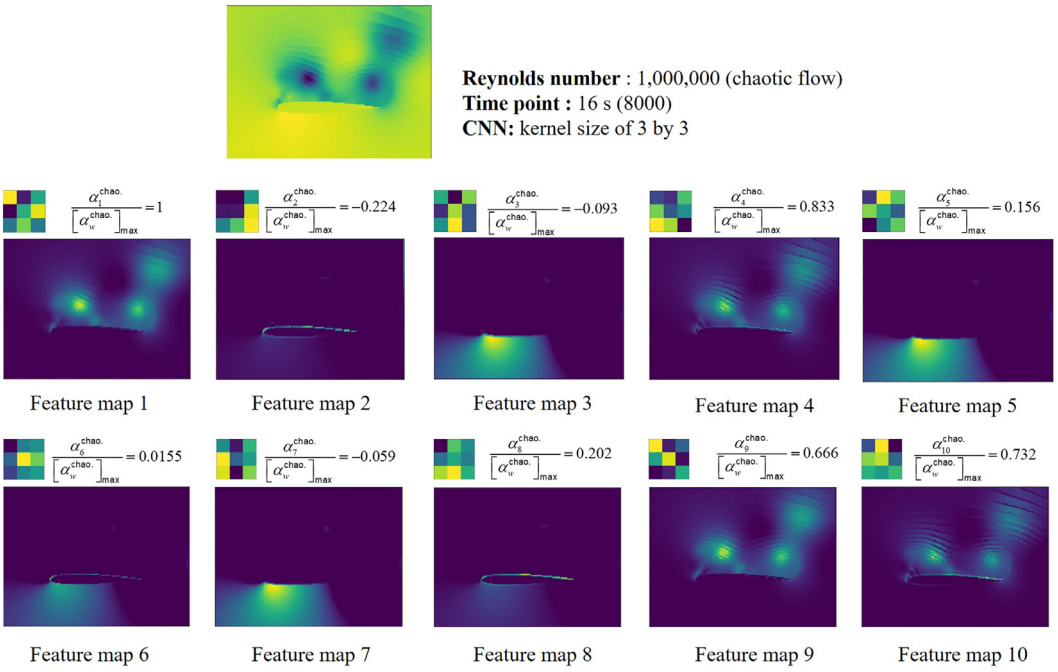

Fig. 4. Ten kernels of the trained CNN and the corresponding feature maps for the example snapshot. The normalized neuron importance weights denotes $\alpha_w^{\text{chao.}} \Big/ \left[\alpha_w^{\text{chao.}}\right]_{\max}$, where the maximum value $\left[\alpha_w^{\text{chao.}}\right]_{\max}$ is $\alpha_1^{\text{chao.}}$.

Table 3
LSTM hyperparameters.

Hyperparameter	Value
Number of units in LSTM	200
Length of sequence (snapshot)	20
Batch size	1
Dropout ratio for LSTM	0.75

Table 4
Selection of training dataset.

Reynolds Number	Sampling region	Flow type
100	8000–8200	Periodic flow
600	8000–8200	Quasi-periodic flow
10,000	8000–8200	Chaotic flow

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. Whats 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 [40]. 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 2.2, 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 4 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 airfoils 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 flows 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 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, Grad-CAM in the Eq. (2) is implemented. Here, we calculated weights of ten feature maps for chaotic flow. The relative weights (also included in Fig. 4) of the feature maps indicate that different certain structures were valued more than others for the emergence

Table 5
Number of functional kernels for different trained CNNs.

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

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 models flow classification. If the bubbles were present much more than was the high-pressure region near the airfoils 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 flows kinematics. For example, Kurtulus [37] 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 5 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 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. 5. 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.

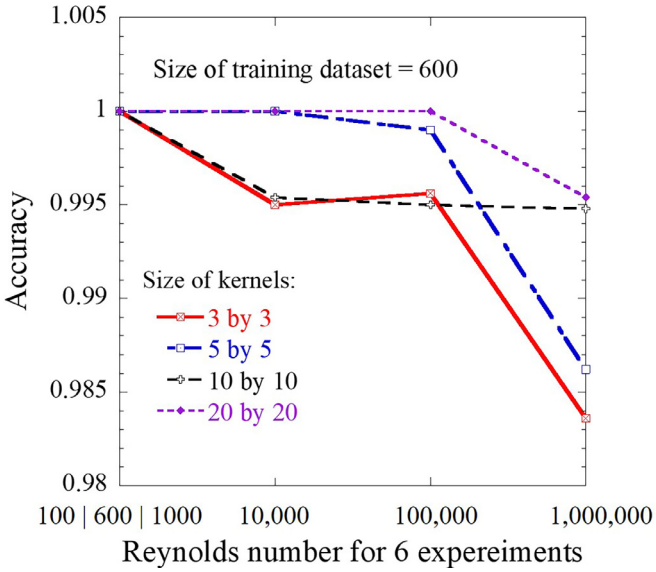

Fig. 5. Test accuracy of 6 different experiments (Reynolds numbers) for 4 CNNs with different kernel sizes.

3.2. Convolutional long-short term memory

In this section, we consider the model with the inclusion of the temporal information. 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 2.1. The resulting classification accuracy was nearly perfect: the machine classified falsely 1 out of 1050 data sequences after 10000 training steps. Figure 6 shows kernels and their corresponding feature maps in the convolutional layer for the trained CNN-LSTM. Grad-CAM is also used to calculate the importance of this feature maps in determining different flow regimes. Compared with those in Fig. 4, it is clear that the CNN also successfully identified three large-scale coherent structures. As was the case with the CNN in Section 3.1, 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 negligible 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.

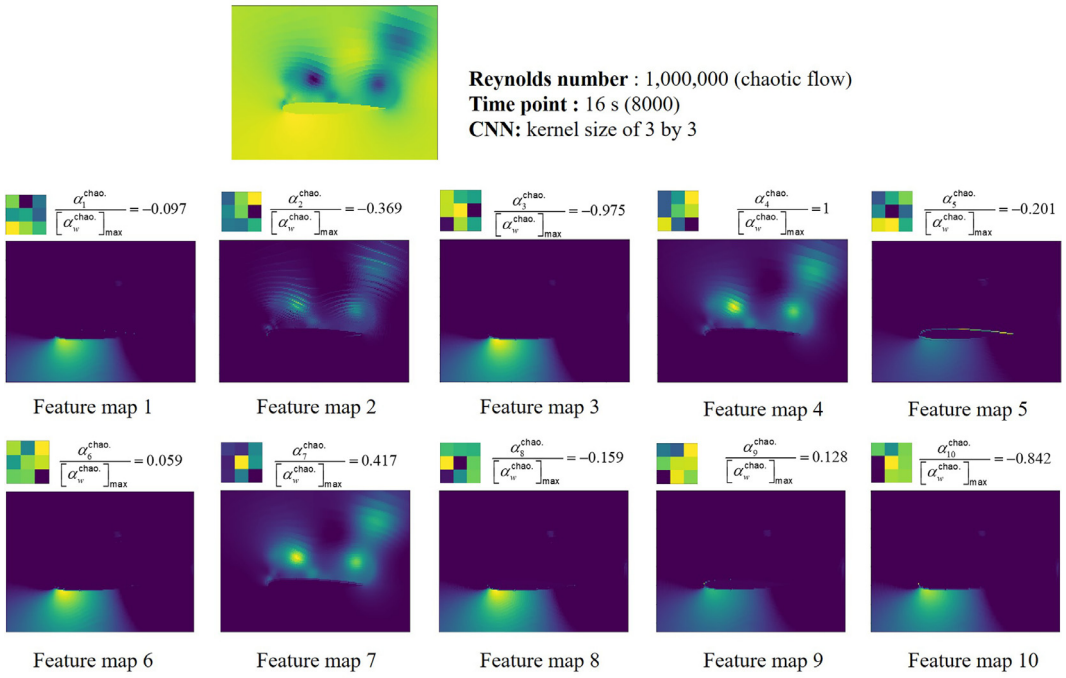

Fig. 6. Ten kernels of the trained CNN-LSTM and the corresponding feature maps for an example snapshot. The number above every feature map is the normalized neuron importance weights.

4. 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, 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 airfoils leading edge. This was accomplished without human intervention or knowledge of the flows kinematics.
2. The presence of localized fluctuations in pressure (bubbles) was found to inform most significantly the models 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 of concern, 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). Moreover, this study, while centered on subsonic flows, outlines a methodology that has potential applicability to transonic buffet scenarios (caused by one or more recompression shocks on the airfoil) as well, paving the way for more efficient and early-stage design evaluations for airfoils, wings, and complete systems. In future work, graph convolutional networks (GCNs) can be considered for providing high resolution for resolving small-scale flow phenomena. And we believe 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. Other flows of engineering interest like jet flows and mixing layers will be studied and other parameters will be varied including the Mach number.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.taml.2023.100482.

References

[1] S. Cai, Z. Mao, Z. Wang, M. Yin, G.E. Karniadakis, Physics-informed neural networks (PINNs) for fluid mechanics: a review, Acta Mech. Sin. (2022) 1–12.
[2] X. Yang, J. Wang, Machine learning in mechanics, Theor. Appl. Mech. Lett. (1) (2022). 100416–100416
[3] S.L. Brunton, B.R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, Annu. Rev. Fluid Mech. 52 (2020) 477–508.
[4] K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence modeling in the age of data, Annu. Rev. Fluid Mech. 51 (2019) 357–377.
[5] J. Kou, W. Zhang, Data-driven modeling for unsteady aerodynamics and aeroelasticity, Prog. Aerosp. Sci. 125 (2021) 100725.
[6] Y. Yin, Z. Shen, Y. Zhang, H. Chen, S. Fu, An iterative data-driven turbulence modeling framework based on Reynolds stress representation, Theor. Appl. Mech. Lett. 12 (5) (2022) 100381.

[7] Z. Li, W. Peng, Z. Yuan, J. Wang, Fourier neural operator approach to large eddy simulation of three-dimensional turbulence, Theor. Appl. Mech. Lett. 12 (6) (2022) 100389.
[8] Y. Li, Z. Wang, W. Jiang, Z. Xie, C. Kong, J. Chang, Research on time sequence prediction of the flow field structure of supersonic cascade channels in wide range based on artificial neural network, Phys. Fluids 34 (1) (2022).
[9] C. Rao, H. Sun, Y. Liu, Physics-informed deep learning for incompressible laminar flows, Theor. Appl. Mech. Lett. 10 (3) (2020) 207–212.
[10] L. Sun, J.-X. Wang, Physics-constrained Bayesian neural network for fluid flow reconstruction with sparse and noisy data, Theor. Appl. Mech. Lett. 10 (3) (2020) 161–169.
[11] J.P. Panda, H.V. Warrior, Evaluation of machine learning algorithms for predictive Reynolds stress transport modeling, Acta Mech. Sin. 38 (4) (2022) 1–13.
[12] L. Zhu, X. Sun, Y. Liu, W. Zhang, One neural network approach for the surrogate turbulence model in transonic flows, Acta Mech. Sin. 38 (3) (2022) 1–14.
[13] Y. Li, J. Chang, Z. Wang, C. Kong, An efficient deep learning framework to reconstruct the flow field sequences of the supersonic cascade channel, Phys. Fluids 33 (5) (2021).
[14] B.D. Tracey, K. Duraisamy, J.J. Alonso, A machine learning strategy to assist turbulence model development, in: 53rd AIAA Aerospace Sciences Meeting, 2015, p. 1287.
[15] K. Duraisamy, Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence, Phys. Rev. Fluids 6 (5) (2021) 050504.
[16] J.-X. Wang, J.-L. Wu, H. Xiao, Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data, Phys. Rev. Fluids 2 (3) (2017) 034603.
[17] J.-L. Wu, J.-X. Wang, H. Xiao, J. Ling, A priori assessment of prediction confidence for data-driven turbulence modeling, Flow Turbul. Combust. 99 (1) (2017) 25–46.
[18] L. Zhu, W. Zhang, J. Kou, Y. Liu, Machine learning methods for turbulence modeling in subsonic flows around airfoils, Phys. Fluids 31 (1) (2019) 015105.
[19] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117.
[20] 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.
[21] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-c. Woo, Convolutional LSTM network: a machine learning approach for precipitation nowcasting, Adv. Neural Inf. Process. Syst. 28 (2015).
[22] S. Ye, Z. Zhang, X. Song, Y. Wang, Y. Chen, C. Huang, A flow feature detection method for modeling pressure distribution around a cylinder in non-uniform flows by using a convolutional neural network, Sci. Rep. 10 (1) (2020) 1–10.
[23] Y. Zhang, W.J. Sung, D.N. Mavris, Application of convolutional neural network to predict airfoil lift coefficient, in: 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2018, p. 1903.
[24] 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, 2016, pp. 481–490.
[25] 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).
[26] B. Colvert, M. Alsalman, E. Kanso, Classifying vortex wakes using neural networks, Bioinspiration Biomimetics 13 (2) (2018) 025003.
[27] D. Tang, E.H. Dowell, Experimental aerodynamic response for an oscillating airfoil in buffeting flow, AIAA J. 52 (6) (2014) 1170–1179.
[28] 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, J. Fluids Eng. 138 (1) (2016).
[29] T. Zhou, S.-S. Feng, E.H. Dowell, Buffeting and lock in of an airfoil at high angle of attack, J. Aircr. 55 (2) (2018) 771–780.
[30] J. Crouch, A. Garbaruk, D. Magidov, Predicting the onset of flow unsteadiness based on global instability, J. Comput. Phys. 224 (2) (2007) 924–940.
[31] K. Bastos, E. Dowell, RANS and DNS study of buffet in subsonic and transonic flow, in: International Forum on Aeroelasticity and Structural Dynamics. IFASD, 2019, pp. 9–13.
[32] K.M.K. Bastos, Computational studies of buffet and fluid-structure interaction in various flow regimes, Duke University, 2020 Ph.D. thesis.
[33] 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, p. 4742.
[34] J. Kühnen, B. Song, D. Scarselli, N.B. Budanur, M. Riedl, A.P. Willis, M. Avila, B. Hof, Destabilizing turbulence in pipe flow, Nat. Phys. 14 (4) (2018) 386–390.
[35] R. Wiebe, L. Virgin, A heuristic method for identifying chaos from frequency content, Chaos 22 (1) (2012) 013136.
[36] E. Rossi, A. Colagrossi, G. Oger, D. Le Touzé, Multiple bifurcations of the flow over stalled airfoils when changing the Reynolds number, J. Fluid Mech. 846 (2018) 356–391.
[37] D.F. Kurtulus, On the unsteady behavior of the flow around NACA 0012 airfoil with steady external conditions at Re = 1000, Int. J. Micro Air Vehicles 7 (3) (2015) 301–326.
[38] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.
[39] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[40] K. Taira, S.L. Brunton, S.T. 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 J. 55 (12) (2017) 4013–4041.