Extraction of Flow Features for Predicting Pressure Distribution Using Convolutional Neural Networks

Prince Tyagi, Priyam Gupta and Raj Kumar Singh

May 17, 2021

Extraction of Flow features for predicting pressure distribution using Convolutional Neural Networks

Prince Tyagi, Priyam Gupta and Raj Kumar Singh

Delhi Technological University, Shahbad Daulatpur, Main Bawana Road, Delhi-110042

Abstract. Recent developments in Artificial Intelligence and Machine Learning have led to new sophisticated approaches to solve complex engineering problems. This study proposes an advanced Convolutional Neural Network based data driven framework to infer load or pressure values from velocity profiles. Convolutional Neural Networks are chosen over conventional Neural Networks because CNNs require reduced training variables which leads to optimized computational time and resources required for training. The model spatially extracts features from fluid flow data using convolution filters and attempts to map them with pressure profile. Hyperparameters of the model are finely tuned to ensure optimum functioning and results. Performance of the predictive model is tested on two flow cases – Converging Diverging Channel and Parametric Bump for multiple Reynolds Numbers. An opensource validated Dataset was utilized for ensuring standardized training of the model. Overall, the framework is found to be effectively efficient and a high degree of accuracy is observed in the results.

Keywords: Fluid Mechanics, Convolutional Neural Network, Machine Learning.

1 Introduction

Calculating loads in a flow field is of significant practical importance in fluid mechanics. Moreover, widely accepted experimental techniques to visualize and measure fluid flow like Particle Image Velocimetry require heavy post-processing to obtain pressure, vorticity values. This demands for an effective and computationally efficient framework which can extract flow field features and map them to desired quantities like pressure, vorticity etc.

The rapid surge in the advancement of deep learning models has enabled the mapping of complex domains with each other. A neural network is widely considered as a universal function approximator with the potential to learn the relationship between involuted domains[1]. The unprecedented advancement in computational power and huge amounts of simulation data has made these deep learning frameworks more powerful than ever before for many engineering problems like active flow control, turbulence modeling, flow field feature extraction and prediction, reduced order models and many more[2]. Convolutional Neural Networks (CNN) are an advanced version of vanilla neural networks with significantly reduced number of trainable weights and ability to extract spatial features from the data [3]. This considerably enhances the computation time and helps the model to extract features from the data more efficiently.

Motivated by the impressive performance of Convolutional Neural Networks in pattern recognition and image analysis [4] this work attempts to develop a CNN based framework to predict loads in a flow field from velocity distribution. The proposed predictive model is tested on two different use cases including a Convergent Divergent Channel and Parametric Bumps for different Reynolds Numbers. The hyperparameters of the model are tuned to optimize prediction accuracy. This framework is evaluated for its potential to optimize computation time and maximize the accuracy of predictions.

The paper has been arranged as follows: Section 2 reviews the existing literature most relevant to the proposed work. Section 3 delineates the methodology followed along with an explanation of fundamental principles related to the work in detail. Section 4 discusses the results obtained and the efficiency of the model. Finally, Section 5 includes the concluding remarks.

2 Literature Review

Machine Learning has been popularly put into practice to solve many problems in fluid mechanics like turbulence modelling, shape optimization, active flow control, and reduced-order models. This is primarily because of these algorithm`s exceptional ability to learn hidden features from the data and provide unprecedented insights into the physical behaviour of the fluid. Wang et al. [5] the improved predictive ability of the Reynolds Averaged Navier Stokes turbulence model by mitigating the discrepancies in the RANS-modeled Reynolds stresses using physics informed machine learning approach. The discrepancies were modelled as a function of the mean flow features. They specifically implemented Random Forest regression technique for this purpose. Beck et al. [6] used CNNs to close Large Eddy Simulation based turbulence models. They developed a non-linear mapping from coarse grid quantities of decaying homogenous isotropic turbulence to the closure terms. Bhatnagar et al. [7] developed a Convolutional Neural Network based framework to predict flow field around airfoils at a different angle of attacks and Reynolds number. Basu et al [8] developed a neural network-based framework to map pressure fields from particle image velocimetry data.

Zhang et al [9] developed a CNN based framework for predicting the lift coefficient of different shaped airfoils for multiple Reynolds Number, Mach numbers and angles of attack. Ye et al [10] used a similar architecture to evaluate the pressure coefficient around the cylinder. They further enhanced their predictions by implementing transfer learning which improved their model`s performance manifold. Murata et al [11] decomposed the flow field using CNN based autoencoder for flow around a cylinder. They showed their model to perform better than the conventional Proper Orthogonal Decomposition Method. Wang et al [12] implemented CNNs for microfluidic applications. They improved the computation time and resources by converting the problem of predicting the fluidic behaviour of microfluidics into an image recognition task. Puffer et al [13] utilized the image feature extraction capability of CNNs to model incompressible fluid flow. Strofer et al [14] developed a data driven framework to differentiate the features of different flows without incorporating physics. Miyanawala et al [15] integrated a reduced order modeling technique with a deep learning framework to predict the unsteady flow fields. These impressive applications and results of Convolutional Neural Networks for problems in fluid mechanics have motivated us to implement them for load prediction from velocity profile to optimize the computation time and enhance the accuracy.

3 Methodology

3.1 Neural Networks

Neural Networks are a network of neurons that contains the neurons connecting each other. These structures are inspired from the biological form of neurons. The simplest form of neural network is called Perceptron which contains simple input to produce an output and has been extensively used to produce outputs for binary classifications. A simple view of a perceptron is shown in Figure 1.

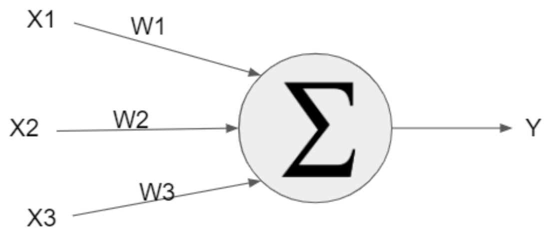


Fig. 1. A perceptron

A perceptron is called a single layered neural network containing input layer, activation layer and an output layer which predicts between range 0 and 1. A general form of the perceptron is a neural network that contains three layers – input, hidden and output layers. The input is transformed by the use of activation functions. Commonly used activation functions are relu, tanh, leaky relu, etc. These structures are also called Multi-layer Perceptron or MLP. Each layer is multiplied with weights and activation function is applied before feeding it to the next layer. The structure is shown in Fig 2.

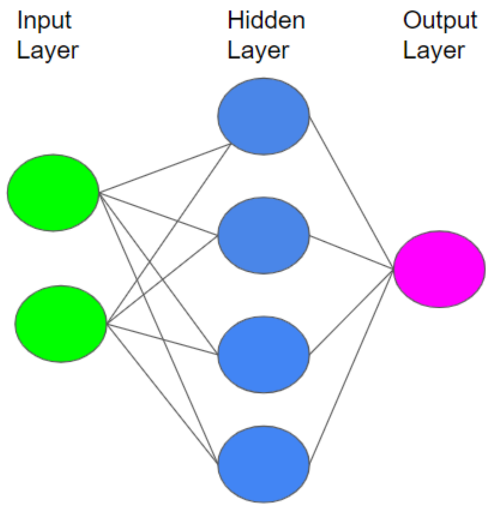


Fig. 2. General structure of a neural network

Mathematically, equation 1 represents the functioning of neural network

$$O = f(b + \sum_{i=1}^{n} x_i \, w_i) \tag{1}$$

where b is the bias, x is the input to the neuron, w is the weight, n is the number of inputs to this layer and i is a counter from 1 to n.

3.2 Convolutional Neural Network

Convolutional Neural Network is a type of neural network which can take the image as input and assigns importance to features based on learnable weights and biases. As compared to other algorithms in machine learning, CNN doesn't need much pre-processing. They are very successful in capturing spatial and temporal dependencies of the image through the use of kernels. The kernels are the first part of these networks. These filters move to the right with certain strides in such a way that it covers the whole part of the image. The objective of these filters is to find out important features from the image. A CNN convolution operation is shown in Fig 3. To reduce the dimensionality in the CNN structures pooling layers are used. There are various types of pooling layers such as max pool, average pool. As denoted by the name, max pool returns the maximum value of pixels covered by the kernel in the image while the average pool returns the mean value. The size of the image so reduced after applying kernels can be kept same or increased by the use of Pooling layers. If I denotes image ,kernel by h, indexes of rows and columns of the resulting feature map matrix F by m and n respectively, then

$$F[m,n] = \sum_j \sum_k h\,[\,j,k\,] \times I\,[\,m-j, n-k\,] \tag{2}$$

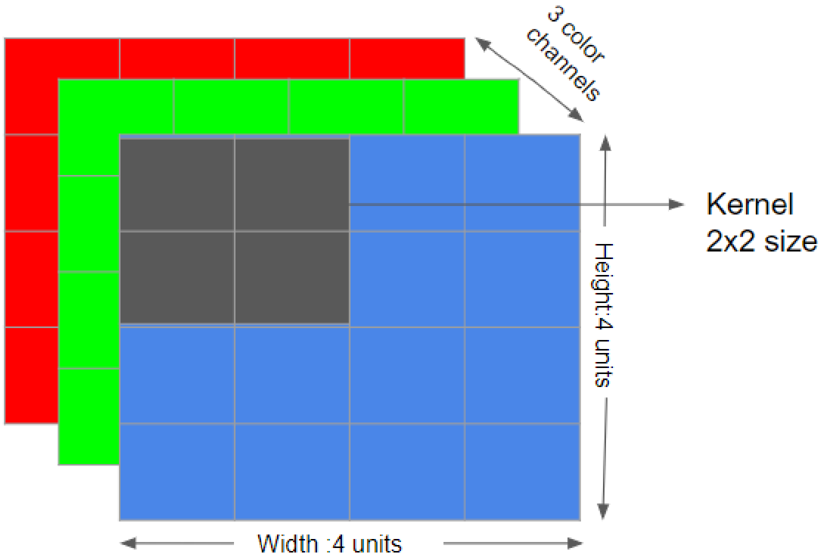
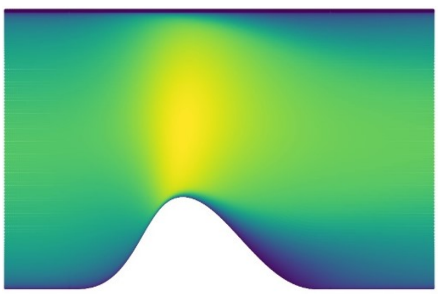


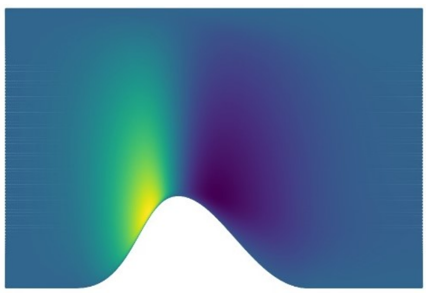
Fig. 3. Convolution Operation in a CNN

3.3 Data Generation

Data is an integral component of the framework proposed in this work. The training data consists of velocity and pressure contours for three different test cases namely, Convergent Divergent Channel, Curved Backward facing step and Parametric Bumps. These profiles have been created using Reynolds Average Navier Stokes Equation based OpenFOAM simulation utilizing k-ε turbulence model. In order to standardize the model performance, we have used open-source validated data developed by McConkey et al [16]. The velocity and pressure contours are converted into images of size 264 X 264 X 3 and standardly normalized before feeding them to the model. A subset of the velocity contour plots for each case are shown in Fig 4.

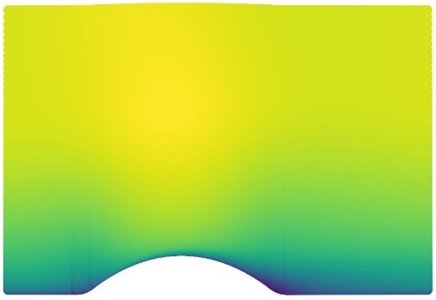


a.1)Velocity in x direction (U_x)

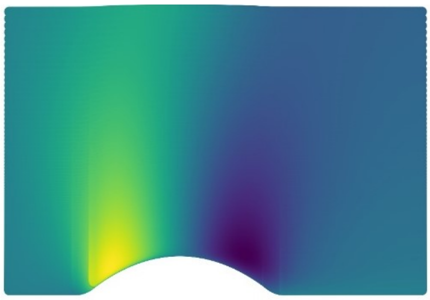


a.2) Velocity in y direction (U_y)

a) Parametric Bump



b.1)Velocity in x direction (U_x)



b.2) Velocity in y direction (U_y)

b) Convergent Divergent Channel

Fig. 4. Sample Velocity Contours from the Dataset

3.4 Supervised Learning Problem

Supervised learning is a type of machine learning problem in which there are output label attached to each of the training and testing data points. If X is the input data point and Y is the output label for each such X, then the algorithm learns a mapping function from input to output. There are some steps needed to be followed in order to perform a supervised learning problem.

1. The foremost step is to decide what kind of data needs to be used for training the algorithm.
2. The next step is to prepare training data. A set of input and output are gathered.
3. In this step, the representation of input data is determined also called feature map.
4. After determining the structure of the input data algorithm is decided.
5. Algorithm is trained on training data and various training parameters are controlled in this step.
6. The accuracy of the algorithm is measured on the testing data in this step.

Our problem is a supervised problem and we have chosen CNN to be trained on our data. The structure of CNN is tabulated in Table 1.

4 Results

The dataset is split into two parts – training and testing in the ratio 80 and 20 respectively. Since the problem is a supervised learning problem, while training the model learned a feature map from input to output. The input images fed to the network are of size 264 x 264 x 3 and are normalized in the range [-1,1].

Table 1. Structure of Convolutional Neural Network

Layer	No of filters	Activation
1st Convolution + Max pooling	2 x 2 , 32 , 3 x 3	relu
2nd Convolution+ Max pooling	2 x 2 ,64 , 3 x 3	relu
3rd Convolution+ Max pooling	2 x 2 ,128 , 3 x 3	relu
4th Convolution+ Max pooling	2 x 2 ,256 , 3 x 3	relu
5th Convolution+ Max pooling	2 x 2 ,512 , 3 x 3	relu
Flatten + 1st Dense	1,000	tanh
2nd Dense	1,000	tanh
Ouptut layer	10,000	linear

The network is trained using an RMS prop optimizer with a learning rate of 2e−3.The tuning of hyperparameters is done by using the grid search method. The loss evolution of the model is shown in Fig 5. As seen in the graph loss is decreasing with epoch and finally converges to value 0.04.

Fig 6 shows the comparison between real and generated samples by the model after training. The quality of the model is assessed by mean squared error (MSE) which is the squared difference between actual and predicted points. Lower value of MSE points indicate towards the closeness of predicted points to actual points. The model is performing well on training and testing data. It is able to predict the points with a MSE of 0.04. From Fig 6. the high proximity of predicted pressure to actual pressure can be visualized. A slight deviation of the predicted values which is observed in the contour plots can be attributed to the small size of training data and relatively smaller size of the network. This can be further improved by training a larger CNN model but it will lead to expensive computation.

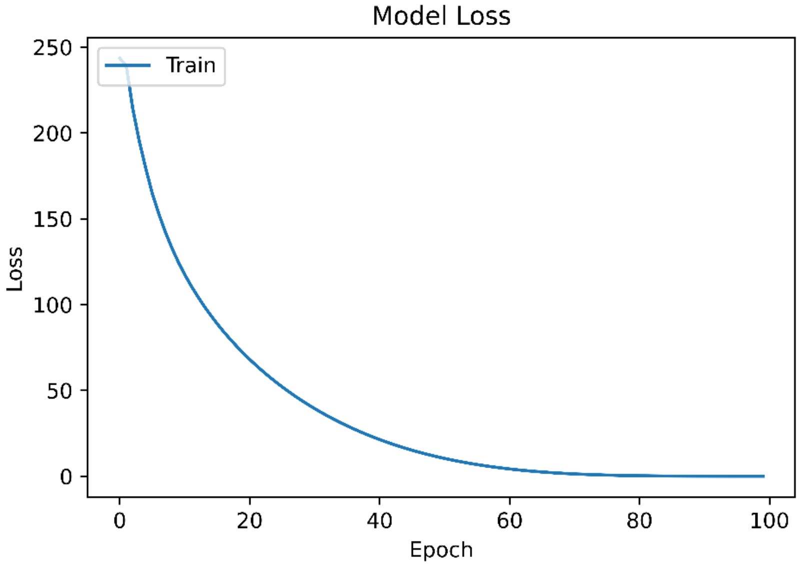
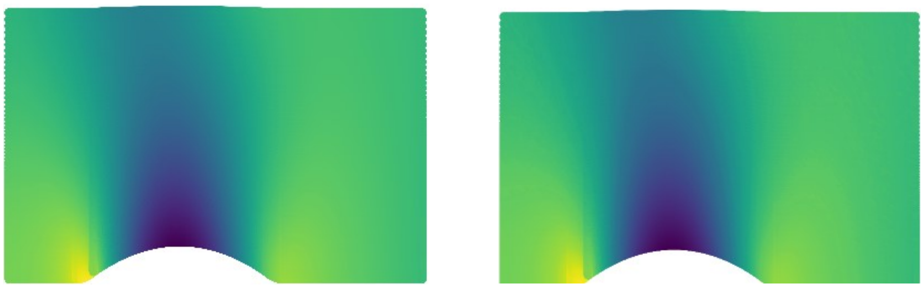
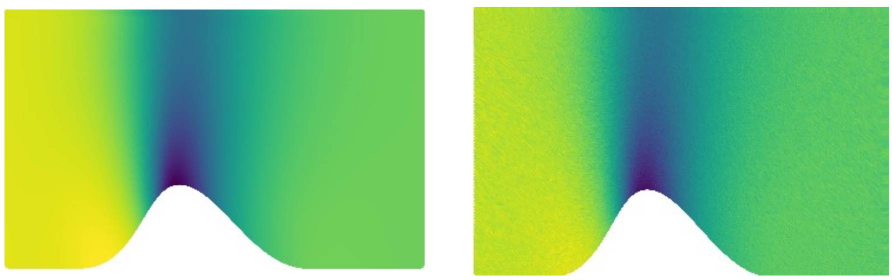


Fig. 5. Loss decreasing as the number of epochs increases



a.1) Actual Pressure Contour a.2) Predicted Pressure Contour

a) Converging Diverging Channel



b.1) Actual Pressure Contour b.2) Predicted Pressure Contour

b) Parametric Bump

Fig. 6. Comparison of actual and generated pressure points. The left side plots are actual pressure point while right side are predicted by the model.

5 Conclusion

In this study, we have predicted pressure points from the velocity field using CNN architecture. The model was trained on 80 data points and performed well with MSE of 0.04. There is very close agreement between predicted and actual pressure value. The slight deviation and noise seen in the result as compared to the actual pressure can be further enhanced by training on more data and using more advanced algorithms like Generative Adversarial Networks.

References

1. Specht, Donald F. "A general regression neural network." *IEEE transactions on neural networks* 2.6 (1991): 568-576.
2. Brunton, S.L., Noack, B.R. and Koumoutsakos, P., 2020. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, *52*, pp.477-508.
3. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*, pp.1097-1105.
4. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. and Chen, T., 2018. Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, pp.354-377.
5. Wang, J.X., Wu, J.L. and Xiao, H., 2017. Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. *Physical Review Fluids*, *2*(3), p.034603.
6. Beck, A., Flad, D. and Munz, C.D., 2019. Deep neural networks for data-driven LES closure models. *Journal of Computational Physics*, *398*, p.108910.
7. Basu, M., Kumar, S., Gupta, P. and Kumar Singh, R., 2020, July. A Quantitative Analysis of Machine Learning Based Regressors for Pressure Reconstruction in Particle Image Velocimetry Applications. In *Fluids Engineering Division Summer Meeting* (Vol. 83716, p. V001T02A016). American Society of Mechanical Engineers.
8. Bhatnagar, S., Afshar, Y., Pan, S., Duraisamy, K. and Kaushik, S., 2019. Prediction of aerodynamic flow fields using convolutional neural networks. *Computational Mechanics*, *64*(2), pp.525-545.
9. Zhang, Y., Sung, W.J. and 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).
10. Ye, S., Wang, Y., Zhang, Z. and Huang, C., 2019, July. Transfer learning for modeling pressure coefficient around cylinder using CNN. In *The 29th international ocean and polar engineering conference*. International Society of Offshore and Polar Engineers.
11. Murata, T., Fukami, K. and Fukagata, K., 2019. Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *arXiv preprint arXiv:1906.04029*.

12. Wang, Junchao, Naiyin Zhang, Jinkai Chen, Guodong Su, Hailong Yao, Tsung-Yi Ho, and Lingling Sun. "Predicting the fluid behavior of random microfluidic mixers using convolutional neural networks." *Lab on a Chip* 21, no. 2 (2021): 296-309.
13. Puffer, F., Tetzlaff, R. and Wolf, D., 1998, April. Modeling almost incompressible fluid flow with CNN. In *1998 Fifth IEEE International Workshop on Cellular Neural Networks and their Applications. Proceedings (Cat. No. 98TH8359)* (pp. 78-82). IEEE.
14. Ströfer, C.M., Wu, J., Xiao, H. and Paterson, E., 2018. Data-driven, physics-based feature extraction from fluid flow fields. *arXiv preprint arXiv:1802.00775*.
15. Miyanawala, T.P. and Jaiman, R.K., 2019, June. A hybrid data-driven deep learning technique for fluid-structure interaction. In *International Conference on Offshore Mechanics and Arctic Engineering* (Vol. 58776, p. V002T08A004). American Society of Mechanical Engineers.
16. McConkey, R., Yee, E. and Lien, F.S., 2021. A curated dataset for data-driven turbulence modelling. *arXiv preprint arXiv:2103.11515*.