COMPUTATIONAL MECHANICS 2019
35th conference with international participation
Srní
November 4 - 6, 2019

Fast pressure prediction along the NACA airfoil using the convolution neural network

O. Bublík[a]

[a] NTIS – New Technologies for the Information Society, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 301 00 Plzeň, Czech Republic

Convolution neural network (CNN) is a class of neural networks developed for image processing. The common use of CNN lies in objects classification or image segmentation, see Fig. 1. A typical CNN architecture consists of a sequence of convolution and pooling layers, see Fig. 1 (left). The convolution layers provide the convolution using a selected kernel function, while the pooling layers reduce the data dimensions. At the output of the CNN is fully connected (dense) neural network, which can make an object classification, see Fig. 1 (left). In the case of image segmentation (Fig. 1 (right)), the additional layers are connected to the output of dense neural network to recover the original shape of the input data. The additional layers are created using convolution and up-sampling (reversed to pooling) layers. The resulting CNN is usually called autoencoder or U-net, see Fig. 2. The autoencoder (U-net) is the base architecture for the use of CNN in physical modelling.

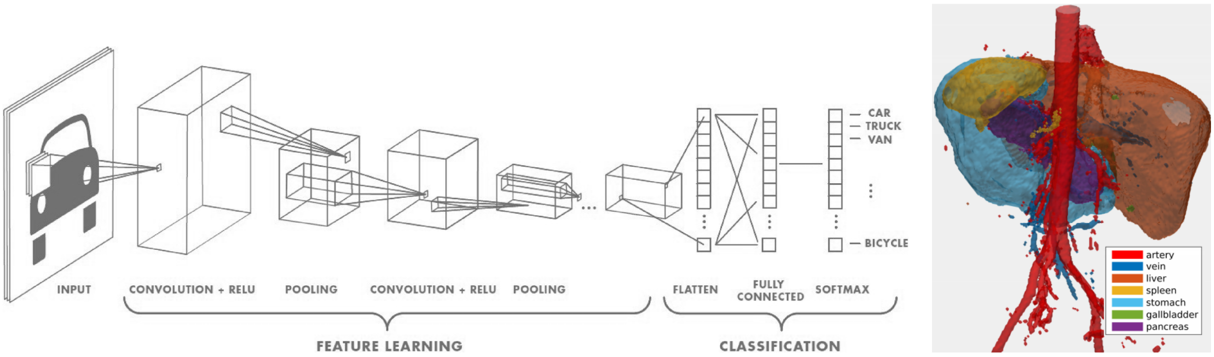

Fig. 1. (*Left*) The use of the CNN for the cars classification. The image was taken from the web site [7]. (*Right*) The use of the CNN for the organ segmentation in the human body [6].

The pioneering use of the CNN for the steady fluid flow simulation was published in the paper [4], where the input image, which contains the boundary information, was transformed to the velocity field. The CNN was trained on the set of solutions, for various boundaries, solved by the lattice Boltzmann method. The results show good accuracy of the generated velocity field. The use of CNN for the unsteady fluid flow simulation was described in [5].

This work aims to use CNN for the pressure prediction along the NACA airfoil and establish the lift and drag coefficients. The knowledge of the lift and drag coefficients in dependence on the angle of attack is the most important airfoil characteristic. The determination of the lift and drag coefficients by the classical CFD methods, such as finite volume or finite element methods, is very time-demanding. This is inappropriate when a new profile is being designed

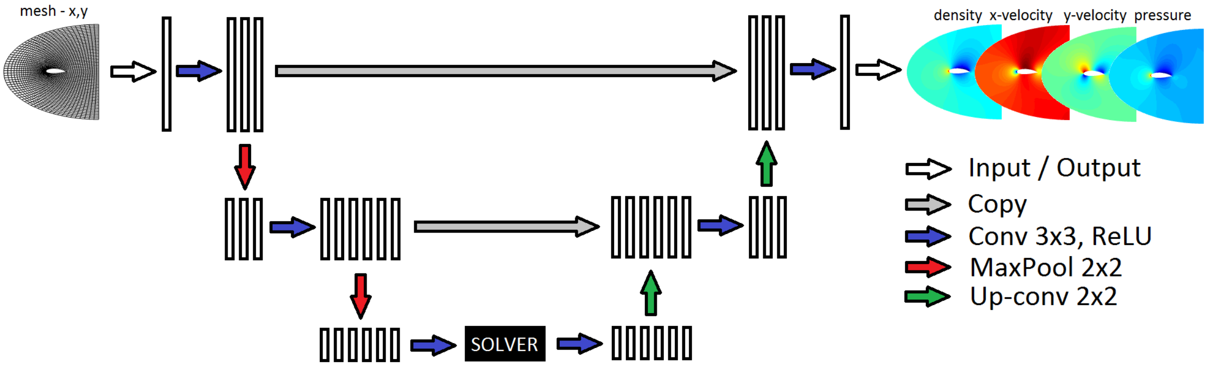

Fig. 2. Architecture of the U-net CNN used for the pressure prediction around a NACA airfoil

using the optimisation techniques, where the calculation of the pressure field for many different variations of the geometry is necessary. The use of CNN aims to utilize the fast prediction of the flow field, from which the pressure distribution around NACA airfoil could be extracted.

For the prediction of the flow field the U-net architecture described above was considered. The *x* and *y* coordinates of the structured C-mesh with 64×32 points was set as a CNN input, see Fig. 3. At the output the flow field, which includes density, pressure and velocity in *x* and *y* directions, is generated. The CNN include two decoding and encoding convolution layers with ReLU (Rectified Linear Unit) activation function. The dense neural network (basically fluid flow solver) between decoder and encoder has four layers. The CNN contains 106 324 trainable parameters in total.

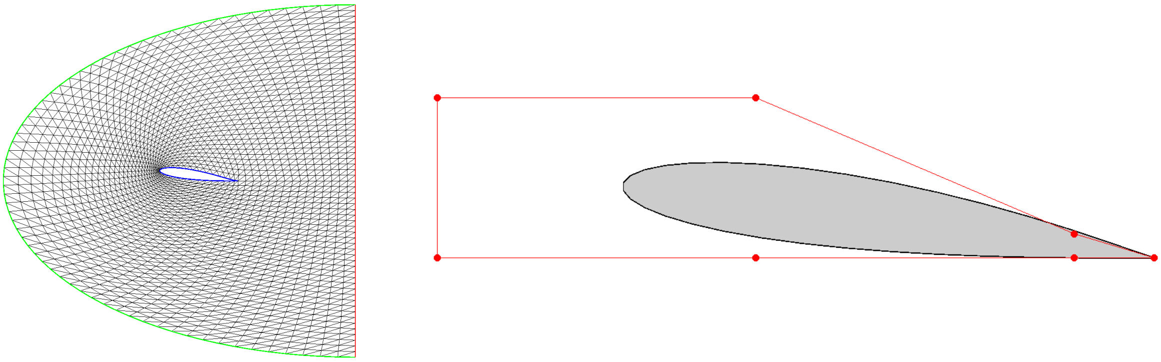

Fig. 3. (*Left*) C-mesh around an airfoil generated using the elliptic mesh generator. (*Right*) Airfoil generated by Bezier curve with eight control points (red dots).

The input C-mesh is generated using the elliptic mesh generator. The airfoil body in the center of the C-mesh is described using the Bezier curves with eight control points $X_p = [x_p, y_p]$, $p = 1, 2, ..., 8$, where the first and the last control points are the same at the tail of the airfoil. For the simplicity, the x_p coordinates of the control points were fixed and y_p coordinates were varying in the interval $[-0.25, 0.25]$. The airfoil was consequently parametrized by six free parameters. The choice of five division per interval for each y_p leads to $6^5 = 7776$ variants of the airfoils. After filtering the inappropriate airfoils, for example with a negative area, the set of 1866 airfoils were left. The simulation of fluid flow around each of airfoils was performed using the open-source CFD software FlowPro [2]. The nonlinear system of Euler equations was chosen for the simulation of inviscid fluid flow. The angle of attack was set to $\alpha = 0$ and Mach number to $M_\infty = 0.4$. For the speed-up of the calculation, only the first order of spatial accuracy was utilised. The numerical simulation of all the 1866 airfoil variants took 7.4 hours of the

Table 1. The relative error of the generated lift coefficient for different angles of attack α and NACA four digit airfoils

α / profile	0012	2412	4412	6615	8607	9210	0020	0030
5°	2.57	3.45	0.24	6.12	2.07	4.76	2.63	2.60
10°	2.89	2.71	2.39	6.29	0.75	0.56	0.93	5.30
20°	4.50	4.11	3.32	3.52	3.88	1.92	3.00	4.54

CPU time (intel i7, 4-cores), where pure computation took 4.5 hours and the mesh generation with the preliminary phase took the rest of the CPU time.

The implementation of CNN was done using the open-source software Keras [3] through the python interface. The Keras software provides an interface to a TensorFlow [1] library, which is an open-source software library for dataflow used for machine learning. The designed neural network was trained on the full set of 1866 airfoils, with RMSProp (Root Mean Square Prop) optimiser. The evaluation of all the airfoils with the CNN took 10.7 s of CPU time, which is approximately 1500 times faster than computation by traditional computation methods.

The precision of CNN generated solution was tested on the set of NACA profiles, which were not presented in the training set. Table 1 shows the relative error between the CFD solution and the generated CNN solution computed as

$$\text{error} = 100\,\frac{|c_L^{CFD} - c_L^{CNN}|}{c_L^{CFD}}\ [\%].$$

The relative error is below five percent in most cases, which is a good result. Together with the speed of generation, CNN can be used as a powerful tool in the optimisation process. Fig. 4 shows the NACA 4412 airfoil and the corresponding CFD and CNN pressure fields around the airfoil body for different angles of attack.

Acknowledgement

This research is supported by the project "Centre of research and experimental development of reliable energy production" TE01020068 of the Technology Agency of the Czech Republic.

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M. et al., TensorFlow: Large-scale machine learning on heterogeneous systems, software, 2015, https://tensorflow.org

[2] Bublík, O., Pecka, A., Vimmr, J., FlowPro – multipurpose CFD software written in Java, Proceedings of the 33rd conference Computational Mechanics, University of West Bohemia, Špičák, 2017, pp. 13-14.

[3] Chollet, F. et al., Keras: The Python deep learning library, 2015, https://keras.io

[4] Guo, X., Li, W., Iorio, F., Convolutional neural networks for steady flow approximation, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, doi: 10.1145/2939672.2939738.

[5] Hennigh, O., Lat-Net: Compressing lattice Boltzmann flow simulations using deep neural networks, 2017, arXiv:1705.09036. (preprint)

[6] Roth, H. R., Oda, H., Zhou, X., Shimizu, N., Yang, Y., Hayashi, Y., Oda, M., Fujiwara, M., Misawa, K., Mori, K., An application of cascaded 3D fully convolutional networks for medical image segmentation, Computerized Medical Imaging and Graphics 66 (2018) 90-99.

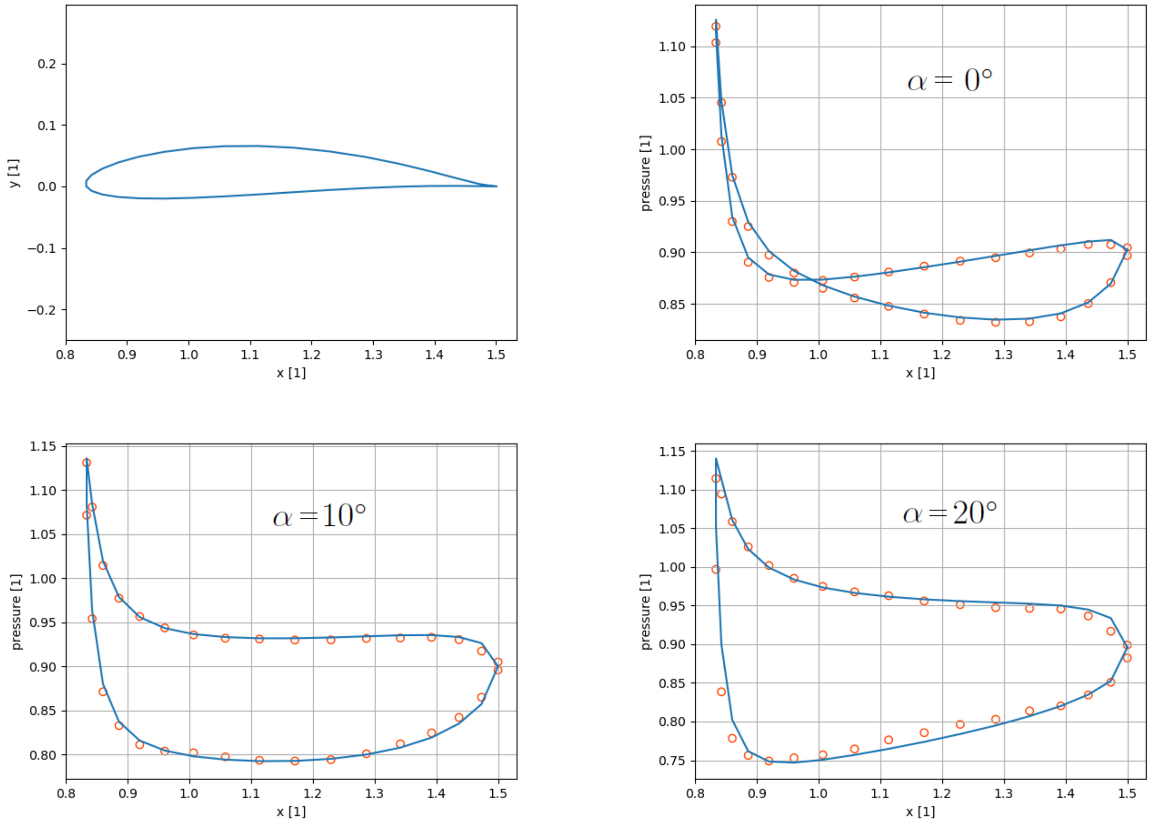

Fig. 4. Pressure field around NACA 4412 airfoil. The blue line represents pressure field obtained by CFD simulation, while red circles represent pressure field generated by CNN.

[7] Saha, S., A comprehensive guide to convolutional neural networks — The ELI5 way, 2018, https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53