Prediction of 2D aerodynamic flow fields on the example of a NACA0012 airfoil - A comparison of experimental data, CFD simulations and CNN predictions

P. Raffeiner[1], M. Berger[2], M. Pillei[2]

[1] Dept. of Medical Technologies, MCI Innsbruck, Austria, p.raffeiner@mci4me.at
[2] Dept. of Environmental, Process & Energy Engineering, MCI Innsbruck, Austria

***Keywords*—2D aerodynamic flow fields, NACA0012, NACA6412, PIV, CFD with RANS, deep learning, comparison.**

***Abstract*—A major benefit of using CNNs for flow stream predictions lies in its ability to significantly reduce computation time. This may be especially valuable for real-time analysis or aerodynamic design, where fast calculations are essential.**

In this work, an established convolutiononal neural network (CNN) from the work of Ribeiro et al. is used as a surrogate model to predict 2D air flow streams. The main objectives of the present work are twofold: first, to develop the turbulent flow prediction capabilities of the CNN, and second, to evaluate the networks predictions by comparing them to traditional computational fluid dynamic (CFD) simulations and experimental particle image velocimetry (PIV) measurements. The evaluation focuses on two test cases: the NACA0012 airfoil at an angle of attack (AoA) of 10° and the NACA6412 airfoil at an AoA of 0°. Both airfoils were inspected when operating at a Reynolds number (Re) of 82,200.

For the CFD simulation, a grid convergence study was conducted to determine the optimal number of mesh elements. Additionally, three different Reynolds-Averaged Navier-Stokes (RANS) turbulence models were analyzed to find the model, that is most suitable to fit the PIV measurements. For the NACA0012 profile, the results of an existing PIV measurement were utilized, while for the NACA6412 airfoil, a new measurement was carried out. In order to predict the flow field around the unseen NACA-airfoils, the CNN was enabled to model turbulent flows. To achieve this, a new set of training samples was generated, incorporating turbulent flows around various types of primitives to train the network.

The comparison of the PIV measurement results and the CFD simulation outcomes for both airfoils show qualitative agreement. However, the experimental data could not be exactly reproduced in both cases.

The comparison of the traditional CFD simulation and the predictions of the network shows initial learning effects in approximating the simulated solution. However, significant differences are still present compared to the ground truth. Several strategies are proposed to enhance the network's prediction ability and address the issue of underfitting as a further step.

I. Introduction

DEEP learning algorithms, in particular convolutional neuronal networks, have the ability to learn and extract features directly from data inputs without the need of manual feature engineering [1]. This potential is used in the present work to demonstrate the utility for the field of fluid dynamics and aerodynamics.

For many years, computational fluid dynamics has been extensively used in a variety of applications ranging from classical industrial and environmental applications to physiological and food processing applications. In those fields, CFD is providing accurate results with great flexibility and cost-effectiveness [2]. However, the accuracy of the solution is strongly influenced by the element size and the number of the mesh elements used in the simulation. Therefore, one beknown limitation of CFD is the time required to solve the highly coupled, non-linear partial differential Navier-Stokes equations [7]. This may be problematic, particularly when fast calculations are needed, such as real-time analysis or during aerodynamic design. Fortunately, the fast-growing field of deep learning (DL) offers a solution.

Numerous recent studies suggest encouraging results by the implementation of machine learning algorithms into CFD. A broad field of applications emerged, ranging from surrogate models [3]–[7] to hybrid-models [8] and super-resolution [9]. One particular study by Dmitrii Kochkov et al. [10], show that the integration of deep learning into CFD has the potential to significantly reduce computation time, with a speedup of 40 - 80 times while maintaining the same level of accuracy as traditional CFD methods.

As is the case in any field of application, the performance of deep learning is heavily dependent on the quality and quantity of the data used to train the model. The field of fluid dynamics shows a significant advantage concerning the training data of the networks. Here, training data can be generated artificially using traditional CFD simulations. While this approach can be computationally and time-intensive, it has the potential to yield significant time savings in the long run. It is important to note that the efficacy of this approach is ultimately limited by the quality and relevance of the generated data.

The objective of this work is to compare and contrast experimental PIV measurement, traditional CFD simulations, and the predictions for the pressure scalar and velocity vector field from a deep learning network using the example of airfoils.

To compare the obtained data from these three methods, the x-component of the velocity vector field is the primary measure that is examined. The attainment of Ribeiro et. al. [3] is utilized for that reason. Their network (DeepCFD) serves as the foundation of this work. To compare the experimental PIV measurement to the prediction of the network is extended to capture turbulent flows.

The results of this comparison will be used to evaluate the potential of deep learning in predicting fluid flow and to assess the differences between experimental measurements, traditional simulations, and deep learning predictions.

However, it is important to note that the objective of this work was not to develop a new CNN or modify the existing CNN in terms of the networks architecture. The only elements modified to yield adequate results in predicting turbulent flows were the training data, as well as the input vector that was altered in consequence. Additionally, some basic hyperparameters such as the batch size and the learning rate, were also adjusted accordingly.

II. Methods

For this work, the NACA0012 and the NACA6412 airfoils from the 4-digit series of NACA airfoils were selected as the primary subject of investigation. The NACA0012 airfoil was particularly chosen based on its well documented characteristics and the availability of experimental data from the work of Sagar Adatrao et al. [11]. The NACA0012 airfoil is a symmetrical airfoil with no chamber and a thickness to cord length ratio of 12%. By contrast, the NACA6412 is an asymmetrical airfoil with a maximum chamber of 6% at 40% of the chord length and a maximum thickness to chord ratio of 12%.

The behavior of both airfoils was tested at a free stream velocity of $10\,\frac{\mathrm{m}}{\mathrm{s}}$. For the NACA0012, the AoA was set to 10° to meet the setup of the experimental data from Sagar Adatrao et al.. The AoA for the NACA6412 was set to 10°. In addition, a PIV measurement was conducted to verify the results of the simulations from the NACA6412 airfoil. The general approach to compare the PIV measurements, the results of the CFD simulation and the prediction of the CNN can be divided into the following steps:

- PIV measurement and evaluation of the obtained data (only for NACA6412)
- Setup and solving of the CFD simulation
- Comparison between experimentally obtained velocity vector field and result from CFD simulation to ensure validity of the simulation
- Generation of primitives that serves as the input vector of the training data for CNN
- CFD simulation of primitives to obtain the output vector of the training data for CNN (pressure scalar field and velocity vector field)
- Training of CNN
- CNN prediction of velocity vector and pressure scalar fields of unseen airfoils (NACA0012 & NACA6412)
- Comparison of CNN prediction with experimental data and CFD simulation results
- Evaluation of flow predictions by CNN and comparison of accuracy with respect to traditional CFD method

A. Particle Image Velocimetry (PIV) Measurement

A PIV measurement was conducted to verify the results of the simulations from the NACA6412 airfoil. Therefore, the NACA6412 airfoil was printed on a Dremel 3D20 3D-printer with a chord width of $100\,\mathrm{mm}$ and an AoA of $0°$. The wing was printed with a layer height of $0.2\,\mathrm{mm}$. The measurement was performed at the in-house wind tunnel of the Management Center Innsbruck. The open-jet open-return wind tunnel has a circular exit cross section with a diameter of $80\,\mathrm{mm}$. To match the simulation, the free stream velocity was set to $10\,\frac{\mathrm{m}}{\mathrm{s}}$. The droplets, consisting of di(2-ethylhexyl) sebacate, were generated by the PivTec GmbH, PIVpart30 seeding generator with a modal diameter of approximately $1.2\,\mathrm{\mu m}$. The Q-switched double pulsed Nd:YAG laser was placed directly in the middle of the chord, parallel to the stream direction. The droplets were illuminated at a frequency of $4\,\mathrm{Hz}$ and a wave length of $532(2)\,\mathrm{nm}$. The puls distance was set to $25\,\mathrm{\mu s}$ with a pulse energy of $220\,\mathrm{mJ}$. The images were recorded with two 2x PCO PCO.2000 (CCD, 14 bit pixel depth, 2048 × 2048 max. pixel resolution) cameras equipped with two Canon EF $50\,\mathrm{mm}$ $f/1.4$ USM objectives. The optical aperture was set to $f_{\#} = 4$. The data set was acquired with 200 double framed images.

The results of the PIV measurement for the NACA0012 airfoil were obtained from Adatrao et al. [11]. For both measurement results, the mean average was determined for the extracted vectors and the x-component of the velocity field was used for the comparison with the simulation results.

B. CFD Simulation

The geometric information of the two-dimensional, symmetric NACA0012 airfoil was obtained in the form of a text file and subsequently imported into the Ansys Fluent computational fluid dynamic software. Here, version 2022 R2 is used. Next, a C-mesh type domain was generated surrounding the airfoil profile and serving as the computational domain. The straight horizontal boundaries of the domain were positioned 20 cord lengths (c) away from the cord line of the airfoil while the straight vertical boundary was located 20c downstream the trailing edge of the airfoil. The semicircular boundary was placed 20c upstream of the trailing edge of the airfoil to complete the domain.

A mesh refinement was performed near the wall and at the wake of the airfoil with gradually increasing elements towards the outlining domain edges. For the mesh, quadrilateral elements were used at the whole domain.

The mesh resolution is highly influencing the time needed to compute a simulation. To reach an optimum between the accuracy of the solution and the computational time, a grid convergence test was conducted. The effect of the number of cells in the domain was measured by the lift coefficient (CL) of the airfoil. Based on the outcome of the grid convergence test, a number around 45800 cells was determined as adequate. Table I shows the results of the grid convergence test.

TABLE I: Results of the grid convergence test for a C-mesh shaped flow domain around the NACA0012 airfoil. The C_L was observed as a measure for the convergence test.

Number of Cells	Lift Coefficient (C_L)
24010	1.004
30400	1.013
45800	1.012
54960	1.012
64120	1.012

The semicircular boundary was assigned as an inlet with a free stream velocity u_∞ of 10 $\frac{\text{m}}{\text{s}}$ at an angle of 10°. The upper and lower boundaries were declared as symmetric wall conditions. The vertical boundary downstream of the chord was defined as a pressure outlet. The boundary condition of the airfoil itself was assigned as a wall property with a no-slip condition.

Furthermore, the solver for the simulation was set up as 2D, double precision, pressure based, steady solver. The incompressibility of the medium at low Mach numbers does not require to solve the energy equations.

The simulation for the NACA6412 airfoil was conducted in a similar manner but for an AoA of 0°.

Due to the major impact of the turbulence model on the solution and the substantial differences between the models, three commonly utilized models were conducted in separate simulations. The realizable/standard (k-ε) [12] and k-ω (SST) [13] are two equation models, whereas the Spalart-Allmaras model [14] is a newer one equation model.

The best results were obtained when using the standard k-ε turbulence model with enhanced wall treatment and the Spalart-Allmaras turbulence model. Therefore, these are used respectively for the simulation of the NACA0012 & NACA6412 profiles and the simulation of the primitives described in II-C.

C. Data Generation

The code to generate the training data for the CNN was written purely in Python and automated for the most part with very little human supervision. The whole process below was written as an asynchronous multiprocess to optimally utilize the performance of the used computer.

Both, the input and output vectors for the CNN are derived by the plain, binary images of the geometry. Here, 8 different primitive shapes were created. The generated primitives consist of circles, rectangles, isosceles triangles, scalene triangles, ellipses, rounded rectangles, pie-slices and chord-slices.

The primitives were created with various heights, widths and AoAs achieved via rotation of the geometries. Three different amounts of training data were generated (763, 2585, 6283) to detect the influence of the number of training points.

With an input/output vector size of 172 × 79, the original dimensions from Ribeiro et al. were preserved. For the purpose of this work, one channel of the input vector was withdraw. Hence, there are only two remaining input channels for the input vector. First is the signed distance field (SDF) of the geometry and second, is the flow region channel (FRC). The SDF is a geometrical representation, based on the distance to the boundary of the object where every pixel contains the distance to the closest edge. The snowy package was used to create the SDFs. For the FRC, four labels were assigned (0 = obstacle, 1 = fluid, 2 = inlet, 3 = outlet) for each pixel. Here, the bottom and left side of the domain were assigned as inlet and the top and right side of the domain as outlet.

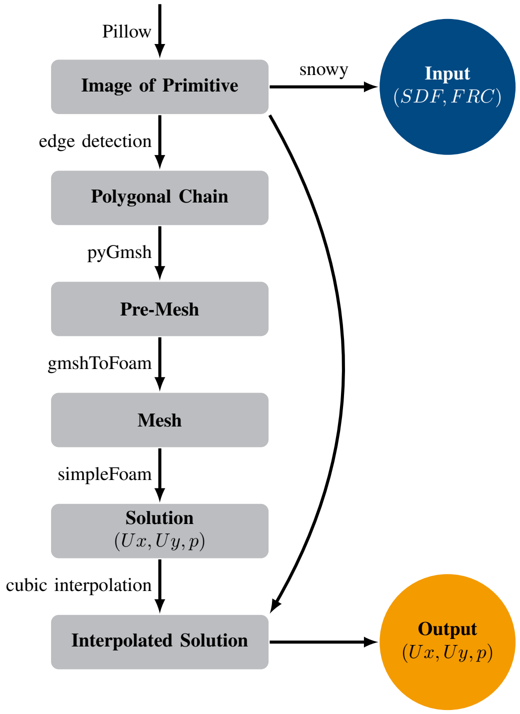


Fig. 1: Flowchart of the data generation process. The rounded rectangles in gray show the steps that were passed to generate the input (Signed Distance Field, Flow Region Channel) and output vectors (velocities in x- and y-direction and pressure). The arrows and the associated text illustrates the methods or packages that were utilized to obtain an outcome. Arrows with no accompanying text should indicate the usage of information from the previous element. The final input and output vectors are represented in the blue and orange circles.

The output vector, which comprises 3 channels (Ux, Uy, p), matches the size of the original vector. To create the output vector a CFD was performed where the mesh was generated using pygmsh and solved via simpleFOAM from the PyFoam package. The inlet velocity was set to 10 $\frac{\text{m}}{\text{s}}$ in x-direction and a simulation time of 1000 s was defined collectively for all simulations.

To interpolate the solution of the highly dense mesh onto the regular grid of 172 × 79, the scipy package was utilized. Here, the cell data of the mesh was extracted in a post-processing step and then interpolated via cubic interpolation.

As a final step, a visual inspection of all results was performed due to the delicate process. Erroneous samples were discarded before entering the data set to the DeepCFD network for training.

The process of data generation for a single sample is illustrated in fig. 1.

D. CNN Training & Predictions

For training of the CNN, the generated samples of 763, 2585 and 6283 datapoints were fed into the DeepCFD network from Ribeiro et al. [3] consecutively. The training was performed

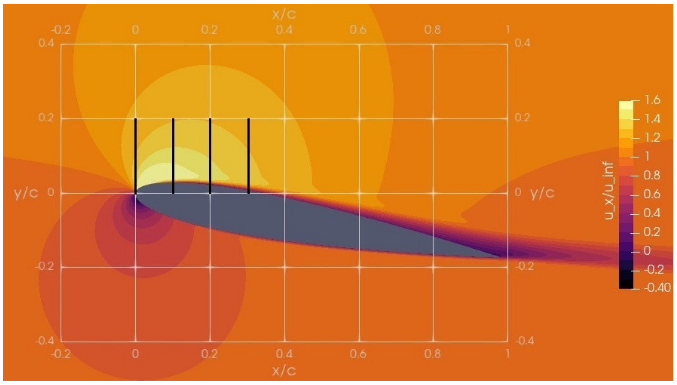


Fig. 2: Results of the flow simulation around the NACA0012 airfoil. The image shows the x-component of the velocity field normalized by the free stream velocity. The coordinates are normalized by the cord length (c) as illustrated by the grid. The black lines represent the vectors along where the line plots were extracted.

on a NVIDIA GeForce RTX 3090. For the training, the batch size was adjusted to a size of 8. The network was trained for 1000 epochs with each of the three data sets.

The data set was split into training and validation data by a ratio of 70% to 30%. It is noteworthy that, the investigated NACA-airfoils are neither part of the training nor of the validation set and therefore completely unseen by the network when a prediction is demanded.

To forecast a result, the best model is saved after the training of 1000 epochs. Then, the single input vector of the unseen NACA-airfoil is fed into this model, generating a prediction by forward propagation.

E. Comparison & Evaluation

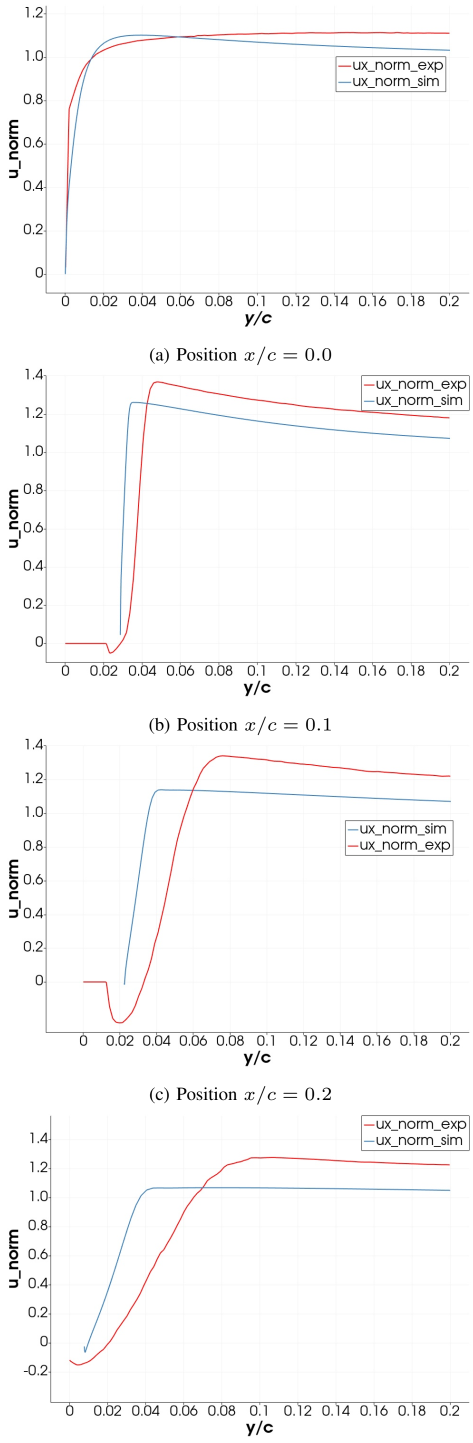
1) PIV Measurement & CFD Simulation: The results of the PIV measurement from Adatrao et al. [11] on the NACA0012 airfoil were examined using the open source visualization software ParaView. Also the simulation results of the NACA6412 airfoil were introduced into ParaView for comparison with the PIV measurement results.

The x-component of the velocity field from the 2D time-averaged flow field was selected for the comparison. A normalization of the velocity field was performed using the free stream velocity $u_\infty = 10\,\frac{\mathrm{m}}{\mathrm{s}}$. The RANS model of the CFD simulation allows to compare the time averaged PIV measurement and the simulation. Here, four positions were selected to extract a line plot along a section. Each section is a vertical line of 0.1 (NACA6412) or 0.2 (NACA0012) chord lengths.

2) CFD Simulation & CNN Prediction: As described in section II-C the output vector of the DeepCFD model is a 3 channel vector (Ux, Uy, p) with a size of 172 × 79 for each channel. All three channels were used for the comparison.

The output from the CFD simulation was (similar to the generation of the output vector in section II-C) interpolated to obtain the regular grid of size 172 × 79. Apparently, all three simulation results (the two velocity fields and the pressure field) were utilized for comparison as well.

Here, each element (pixel) of the 172 × 79 grid from the interpolated CFD simulation is compared to the associated element (pixel) of the predicted output vector from the CNN. The mean absolute error (MAE) is calculated from those two outputs and used as a measure of accuracy. Furthermore, the MAE of all pixels is computed to compare the models trained with a different number of samples.



(a) Position $x/c = 0.0$

(b) Position $x/c = 0.1$

(c) Position $x/c = 0.2$

(d) Position $x/c = 0.3$

Fig. 3: The graphs show the flow velocity normalized by the free stream velocity around the NACA0012 airfoil along a vertical vector from the PIV measurement result (red) and the CFD simulation result (blue). Each graph shows the plot from one of four positions of the vectors. A visualization of the vector locations can be found in fig. 2.

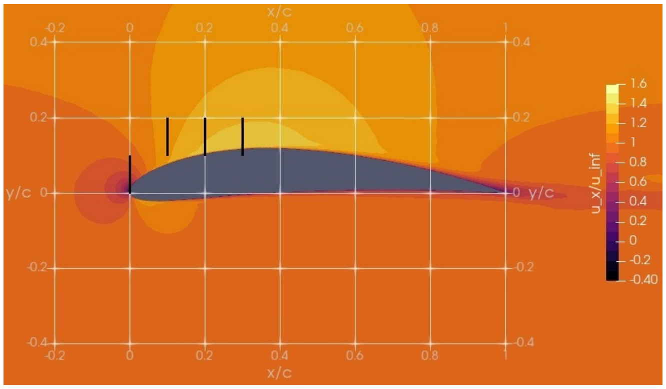
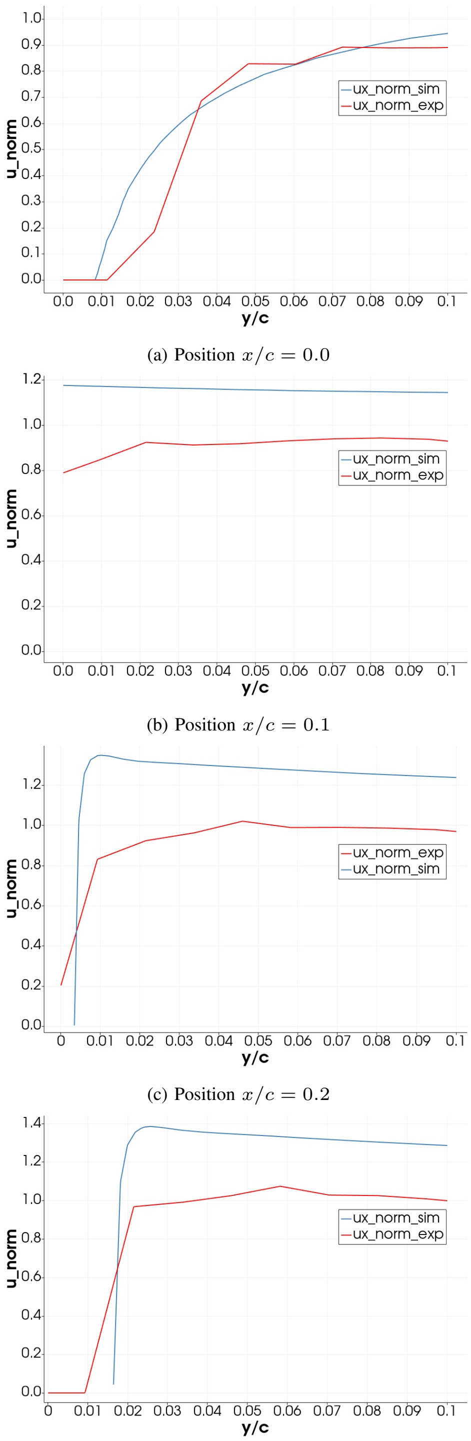


Fig. 4: Results of the flow simulation around the NACA6412 airfoil. The image shows the x-component of the velocity field normalized by the free stream velocity. The coordinates are normalized by the cord length (c) as illustrated by the grid. The black lines represent the vectors along where the line plots were extracted.



(a) Position $x/c = 0.0$

(b) Position $x/c = 0.1$

(c) Position $x/c = 0.2$

(d) Position $x/c = 0.3$

Fig. 5: The graphs show the flow velocity normalized by the free stream velocity around the NACA6412 airfoil along a vertical vector from the PIV measurement result (red) and the CFD simulation result (blue). Each graph shows the plot from one of four positions of the vectors. A visualization of the vector locations can be found in fig. 4.

3) Statistical Evaluation: To verify the results of the CNN prediction, a statistical test was performed. The hypothesis of similarity of the results from the CNN prediction and the CFD simulation (ground truth) was tested. Therefore, the output arrays, which both have a size of 172 × 79 pixels/elements were used in a paired difference test. This is possible due to the prediction of the same result by two different methods (CNN prediction and simulation). The Shapiro-Wilk test could not deliver reliable results due to the exceedance of the maximal sample size by more than factor 2. However, a visual inspection of the array histograms exhibited the distribution was not normally distributed. Therefore, the paired difference, non-parametric, two-tailed Wilcoxon signed-rank test was selected for the analysis with a level of significance (α) equal to 0.05. The test was performed using the wilcoxon function from the scipy package for Python.

III. Results

A. Comparison of CFD Simulation and PIV Results

1) NACA0012: The outcome of the flow simulation around the NACA0012 airfoil is shown in fig. 2. Here, the x-component of the velocity normalized by the free stream velocity is illustrated. The grid illustrated in the figure shows the coordinates normalized by the chord length (c). As described in section II-E1, the results of the PIV measurement and the CFD simulation were analyzed using vertical line plots at four different locations of the airfoil. All four vectors start at a height of 0 and end at the height of 0.2 y/c. The positions in x-direction start at 0 and continue to be 0.1 units apart. These vectors are marked as black lines in fig. 2.

2) NACA6412: The results of the CFD simulation for the NACA6412 chord is illustrated in fig. 4. As for the NACA0012 airfoil, the x-component of the velocity is normalized by the free stream velocity.

The vectors used for the comparison of the CFD simulation results and the experimental PIV measurements for the NACA6412 differ from the vectors used for the NACA0012 due to the different shape of the airfoil and the different AoA. Here, all vectors have a length of 0.1 units. The first vector is formed from the origin, whereas all other vectors originate at a height of 0.1 units in y-direction. In fig. 4 the vectors are represented as black lines.

The results of the analysis is shown in fig. 3. Here, the normalized x-component of the flow velocity from the experiment (in red) and the simulation (in blue) is imaged for every vector.

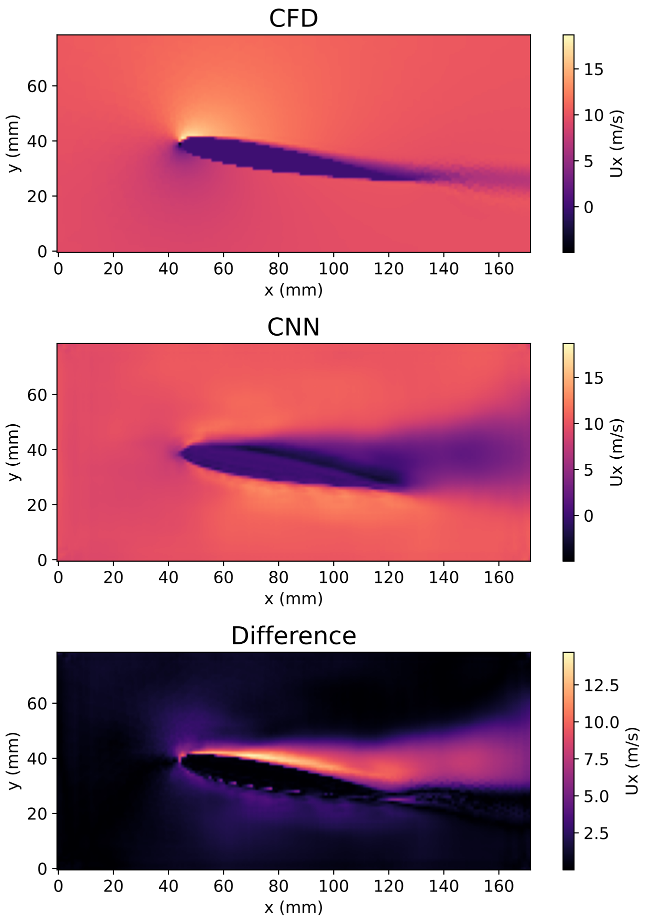


Fig. 6: Comparison for the **NACA0012** airfoil. The figure shows the comparison of the CFD simulation results (top) and the predicted outcome of the CNN (middle) for the flow velocity fields in x-direction (Ux). The bottom image shows the absolute error between the CFD simulation and the CNN prediction. The CNN predictions were attained by a model, trained with **6283** samples.

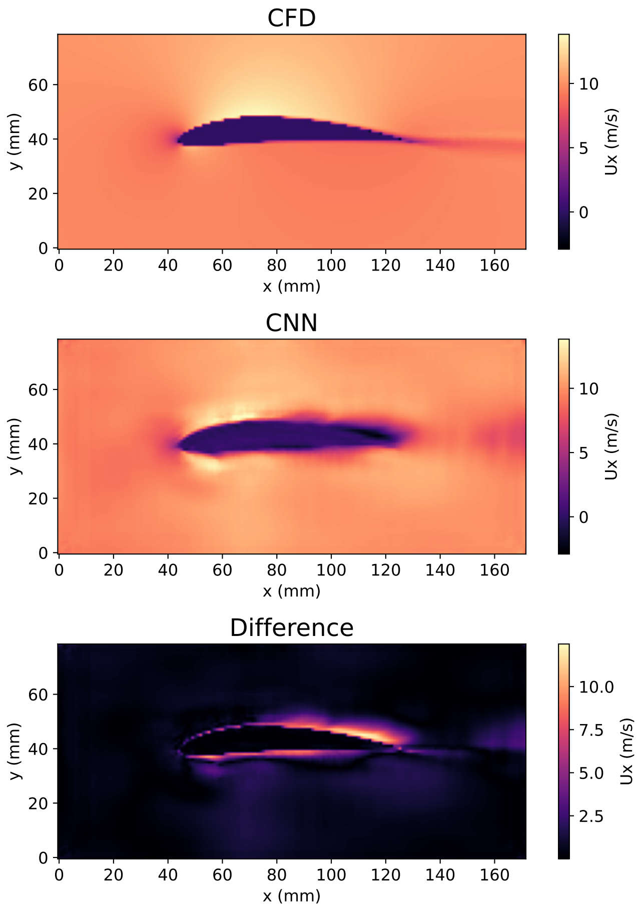


Fig. 7: Comparison for the **NACA6412** airfoil. The figure shows the comparison of the CFD simulation results (top) and the predicted outcome of the CNN (middle) for the flow velocity fields in x-direction (Ux). The bottom image shows the absolute error between the CFD simulation and the CNN prediction. The CNN predictions were attained by a model, trained with **6283** samples.

B. Comparison of CNN Prediction and CFD Simulation Results

The comparison between the CFD results and the CNN predictions is based upon the x-component of the velocity field of the simulation and the CNN. The interpolated result of the CFD simulation serves as the ground truth. As a comparison, the absolute error is calculated between the CFD simulation and the outcome of the CNN for every pixel. The results of this operation can be found in fig. 6 for the NACA0012 airfoil and in fig. 7 representing the NACA6412 airfoil. The predictions of these results were made by the network (DeepCFD) trained with the highest number of training samples. Both figures show the ground truth generated by the CFD simulation in the first column. The middle column represents the prediction made by the CNN with 6283 training samples. The last column illustrates the absolute error calculated by the first two columns.

Furthermore, the MAE is calculated for the whole image instead of a pixelwise operation. Here, the CNN prediction is subtracted from the ground truth (CFD simulation) and the absolute value is calculated for each pixel. Then, the average mean is computed from all pixels. The results can be abstracted from tbl. II for the NACA0012 and tbl. III for the NACA6412 airfoil. Here, all three channels (Ux, Uy, p) were used for comparison.

TABLE II: Mean absolute errors of the CFD simulation results and CNN outputs predicted by models trained with different numbers of samples for the **NACA0012**

Number of Samples	MAE Ux ($\frac{\mathrm{m}}{\mathrm{s}}$)	MAE Uy ($\frac{\mathrm{m}}{\mathrm{s}}$)	MAE p (Pa)
763	2.4704	1.0406	137.9077
2585	3.5930	0.9100	152.6207
6283	8.4229	1.0280	147.3720

TABLE III: Mean absolute errors of the CFD simulation results and CNN outputs predicted by models trained with different numbers of samples for the **NACA6412**

Number of Samples	MAE Ux ($\frac{\mathrm{m}}{\mathrm{s}}$)	MAE Uy ($\frac{\mathrm{m}}{\mathrm{s}}$)	MAE p (Pa)
763	1.5319	0.477 63	79.0899
2585	1.9821	0.4145	99.5374
6283	2.1660	0.5980	94.8296

TABLE IV: Results of the Wilcoxon signed-rank test for the CFD simulation results and CNN outputs predicted by models trained with different numbers of samples for the **NACA0012**

Number of Samples	p-value Ux	p-value Uy	p-value p
763	0.2120	1.3437×10^{-264}	0.0113
2585	5.1205×10^{-264}	4.3481×10^{-226}	0.9393
6283	4.0661×10^{-218}	0.8749	2.4025×10^{-242}

TABLE V: Results of the Wilcoxon signed-rank test for the CFD simulation results and CNN outputs predicted by models trained with different numbers of samples for the **NACA6412**

Number of Samples	p-value Ux	p-value Uy	p-value p
763	0.0494	4.3639×10^{-76}	8.5789×10^{-22}
2585	0.0000	5.8279×10^{-63}	2.9465×10^{-90}
6283	0.002 89	1.3161×10^{-47}	3.7176×10^{-259}

C. Statistical Results

The results of the Wilcoxon signed-rank test for the NACA0012 and NACA6412 can be found in tbl. IV and V respectively. Here, the p-values for the two velocity vector fields and the pressure scalar field are shown, based on the three different number of samples the model was trained with.

IV. Discussion & Future Work

The first two graphs from fig. 3 at the positions 0.0 and 0.1 x/c present satisfactory consensus apart from a minor offset in the far field. At the positions 0.2 and 0.3 x/c it can be observed, that the offset in the far field becomes bigger. Furthermore, the alignment of the results seems to be displaced from the second position (0.1 x/c) on. Due to the late onset of the stream separation in the simulation, there is almost a complete absence of back flow for the examined positions.

Apart from these differences, the results from the simulation correspond with the PIV measurement results for the most part. Thus, the simulation is assessed to be sufficient keeping in mind, that the objective of the simulation was not to utterly mimic the PIV measurement results but to serve as a reference for all further simulations of the primitives.

The simulation results from the NACA6412 airfoil illustrated in fig. 4, show a thin and steep gradient around the airfoil which flattens out at the tip and the trailing edge of the airfoil. The stream does not detach at any point from the chord. Thus, no back flow areas are present. The suction side shows a characteristic increase of up to 1.4 of times the free stream velocity, whereas the pressure side of the airfoil remains almost continuous throughout the whole chord width. The results from the PIV measurement, that was additionally conducted, show qualitative consensus with the simulation. The graphs along the vectors at the positions x/c = 0.1, 0.2 and 0.3 all show an offset of the same size. This is can be explained due to the influence of the wakes originating from the wind tunnels upper boundary. The graph at position x/c = 0.0 does not exhibit this issue because the corresponding vector is at a different height (y/c = 0.0 - 0.1) compared tho the remaining vectors (y/c = 0.1 - 0.2).

When reviewing the results from comparison of the NACA0012 simulation and CNN prediction summarized in tbl. II, it becomes apparent, that the MAE is within the same order as the maximum magnitude of the velocities simulated via CFD for both the x- and y-direction. Also, the MAE of the pressure field is within the same order as the maximum magnitude of pressure. These modest results, yield to the opinion of the networks unability to accurately predict turbulent flow fields from the current setup. Furthermore, when observing the results of the MAE from the table, the number of training samples does not exhibit any influence on the outcome of the CNNs prediction.

The same is true for the comparison of the results between the CFD simulation and the CNN prediction of the NACA6412, summarized in tbl. III. Here, the order of the MAE is also within the same range as the magnitude of the maximum values from the velocity fields and the pressure field. Additionally, the number of training samples does neither have a positive nor a negative effect on the MAE for all three fields. Although, the results of the MAE do not indicate an improvement regarding the different numbers of training samples overall, local advances in the prediction of certain characteristics such as the slower flow in the turbulent wake area, downstream of the airfoil, the formation of pressure gradients around the airfoil and the representation of the airfoil itself can be observed when inspected visually.

To put this into perspective, the original network introduced in the work of Ribeiro et al. [3] reached a accuracy of less than 10% error for most values. However, the utilization of the network in this context was for non-uniform steady laminar flows. The prediction of turbulent flows may be comparatively more difficult to predict.

Additionally, the work of Bhatnagar et al. [15] proposes a MAE level of less than 10% over the entire flow field for turbulent flows. Here, only three different airfoils (S805, S809, S814) were used as training samples with different AoA and Re. This resulted in a total of just 252 samples where 85% used as the training set and the remaining data sets are utilized for validation. Furthermore, the network was trained for 30,000 epochs. Three airfoils were considered as unseen geometries to further explore the abilities of the network. Here, a "new airfoil" was created from an averaged shape of the S809 and S814 airfoils. The other two profiles were the S807 and S819 airfoil. Moreover, it must be noted, that the shape of all three unseen chords more or less lies within the shapes of the trained airfoils making the prediction of the unseen airfoils similar to the prediction of the validation or training chords.

Another point of reference is the work from Thuerey et al. [16], that yields an accuracy of the mean relative pressure and velocity error of less than 3% for their best models. Their CNN was trained for 1200 epochs with RANS simulated samples of flows around 1505 different airfoils at various AoAs and Res. This resulted in a total of 26,722 target samples.

Several approaches occur to improve the networks capacity and to prevent the present underfitting issue:

- The plainest strategy would be to increase the training duration. The network was trained for merely 1000 epochs. As discussed, the model of Bhatnagar et al. [15] was trained for 30,000 epochs.

- Secondly, even though the number of training samples only showed a minor improvement on the prediction capabilities, a drastic increase could anyway lead to better outcomes. For example, Thuerey et al. [16] used a total of 26,722 samples for training.
- Moreover, the enhancement of the training data quality in regards to the number and distribution of different shapes may also lead to success.
- Also, the hyperparameters can be adjusted to optimize training and the prediction output.
- Finally, the network architecture has a big influence on the results. The number of layers and the filter sizes can be increased to take countermeasures against underfitting.

Apparently, a combination of the above mentioned strategies might also be successful.

V. Conclusion

The objective of this work was twofoldly. (1) Enlarging the capabilities of the DeepCFD network to predict turbulent flows rater than just non-uniform steady state laminar flows as in the original configuration. (2) The comparison of the predictions with respect to real applications by verifying the outcome of the simulations that are used as the ground truth.

The second objective was addressed first, to ensure the validity of the simulation process for the primitives which then later serve as the output vector for the network. For that reason, the data from the PIV measurement of the NACA0012 airfoil was used as a reference point and compared to the outcome of the simulation. Although, the simulation results did not entirely meet the results from the PIV measurement, the simulation process was still regarded to be adequate.

To address the first objective, the original input and output vectors of the network were emulated for turbulent streams. The trained network was then used to predict the velocity fields and the pressure field around the unseen NACA0012 and NACA6412 airfoil. Unfortunately, the results of those predictions were inaccurate with an MAE of the same order as the investigated flows. Furthermore, the Wilcoxon signed-rank test also disclosed low conformance compared to the simulation results. Due to this underfitting issue, the original idea of extending the image resolution and to incorporating different free stream velocities were discarded.

However, the issue of underfitting may be addressed through different approaches, in the future. For example by increasing the training time, the number of samples or the networks layer depth or filter sizes. These strategies all yielded success for other works [15], [16].

Even though, alike studies regarding DL predictions of flow fields exist, most of them do either not cover turbulent flows or are specialized to certain geometries such as airfoils. With this work, it should be possible to (1) predict turbulent flow fields and (2) predict flow fields for a broad variety of obstacle shapes.

Acknowledgment

The author extends heartfelt thanks to Manuel Berger, PhD, for his exceptional supervision and guidance throughout this research, with his valuable insights being instrumental in shaping this work. Additionally, appreciation is expressed to Prof. Dr.-Ing. Martin Pillei for the support provided in the laboratory, which significantly enhanced the quality of this study.

References

[1] M. Bojarski *et al.*, "End to end learning for self-driving cars," 2016.

[2] B. Xia and D.-W. Sun, "Applications of computational fluid dynamics (cfd) in the food industry: a review," *Computers and Electronics in Agriculture*, vol. 34, no. 1, pp. 5–24, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169901001776

[3] M. D. Ribeiro, A. Rehman, S. Ahmed, and A. Dengel, "Deepcfd: Efficient steady-state laminar flow approximation with deep convolutional neural networks," 2021.

[4] H. Kang *et al.*, "A new fluid flow approximation method using a vision transformer and a U-shaped convolutional neural network," *AIP Advances*, vol. 13, no. 2, 02 2023, 025233. [Online]. Available: https://doi.org/10.1063/5.0138515

[5] S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik, "Prediction of aerodynamic flow fields using convolutional neural networks," *Computational Mechanics*, vol. 64, pp. 525–545, 2019.

[6] X. Guo, W. Li, and F. Iorio, "Convolutional neural networks for steady flow approximation," 08 2016, pp. 481–490.

[7] O. Obiols-Sales, A. Vishnu, N. Malaya, and A. Chandramowliswharan, "Cfdnet: A deep learning-based accelerator for fluid simulations," in *Proceedings of the 34th ACM international conference on supercomputing*, 2020, pp. 1–12.

[8] A. Usman, M. Rafiq, M. Saeed, A. Nauman, A. Almqvist, and M. Liwicki, "Machine learning computational fluid dynamics," in *2021 Swedish Artificial Intelligence Society Workshop (SAIS)*, 2021, pp. 1–4.

[9] K. Fukami, K. Fukagata, and K. Taira, "Super-resolution reconstruction of turbulent flows with machine learning," *Journal of Fluid Mechanics*, vol. 870, pp. 106–120, 2019.

[10] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer, "Machine learning–accelerated computational fluid dynamics," *Proceedings of the National Academy of Sciences*, vol. 118, no. 21, p. e2101784118, 2021. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.2101784118

[11] S. Adatrao, M. Bertone, and A. Sciacchitano, "Multi-δt approach for peak-locking error correction and uncertainty quantification in piv," *Measurement Science and Technology*, vol. 32, no. 5, p. 054003, 2021.

[12] P. Huang, J. Bardina, and T. Coakley, "Turbulence modeling validation, testing, and development," *NASA technical memorandum*, vol. 110446, pp. 10–2514, 1997.

[13] F. Menter, *Zonal Two Equation k-w Turbulence Models For Aerodynamic Flows*, 1993. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.1993-2906

[14] P. Spalart and S. Allmaras, "A one-equation turbulence model for aerodynamic flows," in *30th aerospace sciences meeting and exhibit*, 1992, p. 439.

[15] S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik, "Prediction of aerodynamic flow fields using convolutional neural networks," *Computational Mechanics*, vol. 64, pp. 525–545, 2019.

[16] N. Thuerey, K. Weißenow, L. Prantl, and X. Hu, "Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows," *AIAA Journal*, vol. 58, no. 1, pp. 25–36, jan 2020. [Online]. Available: https://doi.org/10.2514%2F1.j058291