mathematics

MDPI

Article

Alternative Artificial Neural Network Structures for Turbulent Flow Velocity Field Prediction

Koldo Portal-Porras [1], Unai Fernandez-Gamiz [1,*], Ainara Ugarte-Anero [1], Ekaitz Zulueta [2] and Asier Zulueta [1]

[1] Nuclear Engineering and Fluid Mechanics Department, University of the Basque Country, UPV/EHU, Nieves Cano 12, Vitoria-Gasteiz, 01006 Araba, Spain; koldo.portal@ehu.eus (K.P.-P.); augarte060@ikasle.ehu.eus (A.U.-A.); azulueta@arrasate.eus (A.Z.)

[2] System Engineering and Automation Control Department, University of the Basque Country, UPV/EHU, Nieves Cano 12, Vitoria-Gasteiz, 01006 Araba, Spain; ekaitz.zulueta@ehu.eus

* Correspondence: unai.fernandez@ehu.eus

Abstract: Turbulence in fluids has been a popular research topic for many years due to its influence on a wide range of applications. Computational Fluid Dynamics (CFD) tools are able to provide plenty of information about this phenomenon, but their computational cost often makes the use of these tools unfeasible. For that reason, in recent years, turbulence modelling using Artificial Neural Networks (ANNs) is becoming increasingly popular. These networks typically calculate directly the desired magnitude, having input information about the computational domain. In this paper, a Convolutional Neural Network (CNN) for predicting different magnitudes of turbulent flows around different geometries by approximating the equations of the Reynolds-Averaged Navier-Stokes (RANS)-based realizable k-ε two-layer turbulence model is proposed. Using that CNN, alternative network structures are proposed to predict the velocity fields of a turbulent flow around different geometries on a rectangular channel, with a preliminary stage to predict pressure and vorticity fields before calculating the velocity fields, and the obtained results are compared with the ones obtained with the basic structure. The results demonstrate that the proposed structures clearly outperform the basic one, especially when the flow becomes uncertain. In addition, considering the results, the best network configuration is proposed. That network is tested with a domain with multiple geometries and a domain with a narrowing of the channel, which are domains with different conditions from the training ones, showing fairly accurate predictions.

Keywords: Deep Learning (DL); Computational Fluid Dynamics (CFD); Artificial Neural Network (ANN); Convolutional Neural Network (CNN); turbulent flow

Citation: Portal-Porras, K.; Fernandez-Gamiz, U.; Ugarte-Anero, A.; Zulueta, E.; Zulueta, A. Alternative Artificial Neural Network Structures for Turbulent Flow Velocity Field Prediction. *Mathematics* **2021**, *9*, 1939. https://doi.org/10.3390/math9161939

Academic Editors: Maria Luminița Scutaru and Efstratios Tzirtzilakis

Received: 15 June 2021
Accepted: 12 August 2021
Published: 14 August 2021

1. Introduction

For many years, turbulence in fluids has been a popular research topic due to its impact on a wide variety of applications. Several experimental studies of turbulent flows have improved the understanding of turbulent behaviour and have been used to design more efficient systems. Even though these experiments have been very valuable, there are cases where experimentation is either too expensive or impractical. In these cases, CFD provides a more detailed insight into the physics of turbulent flows. Although CFD has the potential to predict accurately the behaviour of flows, decreasing the need for conducting experiments, it has two main disadvantages. The first disadvantage is the high computational cost of the simulations, which can be prohibitive in cases with very complex geometries or in cases where very accurate turbulence models, such as LES (Large Eddy Simulation) or DNS (Direct Numerical Simulation), are required. The second disadvantage is the influence of the user, especially in the generation of the mesh and the selection of the closure model. These problems, coupled with the growth of artificial intelligence, have led to an increasing number of studies using Deep Learning (DL) techniques applied to CFD, either as a complement to the simulations or to perform the simulations directly.

Numerous authors have used DL techniques to complement numerical simulations performed using CFD. Ray and Hesthaven [1] designed an ANN to detect cells where there is a discontinuity in the results. Liu et al. [2] established a method based on Deep Metric Learning (DML) to determine the optimal time-step value in non-stationary simulations. Bao et al. [3] applied a physically driven approach to improve the modelling and simulation capability of a coarse mesh, and Hanna et al. [4] designed a DL algorithm to predict and decrease the error of the results obtained on a coarse mesh. By using coarser meshes, substantial reductions in computational cost were obtained in both studies.

Several authors oriented DL methods to the analysed geometry. For example, Yan et al. [5], Zhang et al. [6], and Tao and Sun [7] improved the performance and efficiency of various geometries using DL techniques, and reached aerodynamic optimization.

However, the goal of the vast majority of studies using DL techniques applied to CFD is to obtain fluid characteristics. Guo et al. [8] applied a CNN to achieve slightly inaccurate but very fast predictions of stationary flow fields around solid objects. Ling et al. [9] used a Deep Neural Network (DNN) to model Reynolds stress tensors with Reynolds-Averaged Navier-Stokes (RANS) turbulence modelling, achieving a remarkable improvement of the results obtained in CFD simulations. Lee and You [10] predicted the shedding of non-stationary laminar vortices on a circular cylinder using a Generative Adversarial Network (GAN), focusing on explaining the learning potential of the solution of the Navier-Stokes equations. Liu et al. [11] and Deng et al. [12] designed impact and vortex detection methods, respectively, using CNN-based techniques.

Ribeiro et al. [13] and Kashefi et al. [14], using CNN architectures, achieved very accurate results for velocity and pressure fields of stationary fluids around simple shaped obstacles, with a computational cost three to five orders of magnitude lower than CFD simulations. In addition, in the study conducted by Kashefi et al. [14], different velocity and pressure fields were obtained with slight modifications of the geometry, which is essential for design optimization.

Among the previously mentioned studies, the study of Guo et al. [8] is the only one in which three-dimensional domains are analysed, the rest of them only analysing two-dimensional domains. Nowruzi et al. [15] analysed the behaviour of two airfoils in 2D and 3D using CFD and an ANN, and showed good agreements between the results obtained by both methods. Compared to two-dimensional systems, the main disadvantage of analysing three-dimensional systems using DL is the limited workspace [14]. For this reason, Mohan et al. [16] developed a DL-based infrastructure that performs a dimensional reduction of the geometry in order to analyse the flow characteristics susequently.

Although most studies are focused on laminar flows, there are several studies where turbulent flows are examined. Fang et al. [17] applied DL techniques for turbulent channel flow predictions, and Thuerey et al. [18] created a CNN to approximate the velocity and pressure fields of the RANS-based Spalart-Allmaras turbulence model on airfoils.

All the aforementioned studies have required prior CFD simulations to train the ANN. However, Sun et al. [19] designed a structured DNN architecture to approximate the solutions of the parametric Navier-Stokes equations. Instead of using data obtained from simulations, this DNN is trained by minimising only the error of the mass and momentum conservation laws of the flows, thus avoiding the computational expense of CFD simulations. Nonetheless, their study shows that data-driven ANNs are more accurate than this kind of network.

This paper aims to compare the basic network structure for velocity field prediction with alternative network structures, which include a previous stage to calculate pressure and vorticity fields, providing more information about the flow to the network. The remainder of the manuscript is divided as follows: Section 2 explains the methodology followed to conduct CFD simulations, designing the CNN and the different neural network structures and training them; Section 3 displays and compares the results obtained with the proposed different structures; and Section 4 shows an evaluation of the ability of

the proposed neural network to make predictions under different conditions from the training ones.

2. Methodology

2.1. CFD Setup

Numerical simulations by means of CFD were conducted to obtain the required velocity, pressure, and vorticity fields for training, validating, and testing the studied CNNs. To perform these simulations, the Star-CCM+ [20] CFD commercial code was used.

The numerical domain consists of a two-dimensional 128 × 256 mm plate, with a geometry located on its geometrical center. The left and right sides of the plate are set as inlet and outlet, respectively, whereas top and bottom sides and the geometry contour are set as walls with no-slip conditions. A detailed view of the numerical domain is provided in Figure 1.

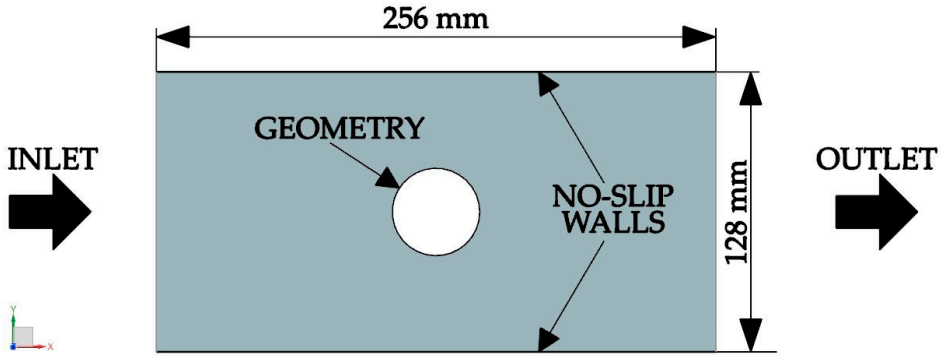


Figure 1. Numerical domain.

In order to collect enough samples for training the network, a total of 2065 simulations were performed. Each simulation was carried out with one of the geometries shown in Table 1. These geometries are based on the geometries of the study of Kashefi et al. [14], and they were generated by changing the size and orientation of eight basic geometries.

With the previously mentioned domain, an unstructured polygonal mesh of around 50,000 cells was generated. This mesh contains more cells near the boundaries in order to ensure good results on the most critical areas. An example of the used meshes can be shown in Figure 2.

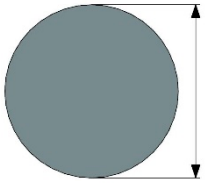
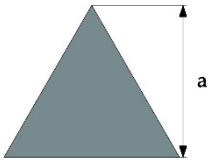
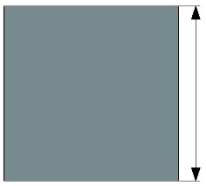
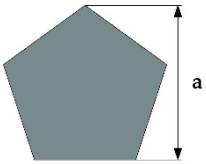
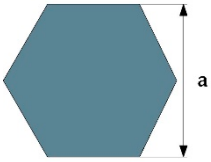
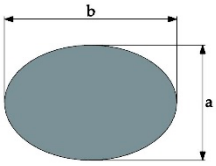
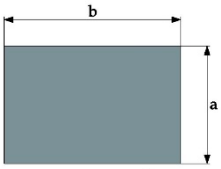
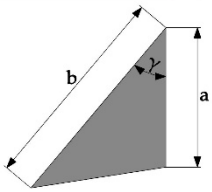
To verify sufficient mesh resolution of the generated meshes, the General Richardson Extrapolation method [21] was performed, applied to the drag coefficient. For this study, the case of a circle with a = 0.02 m is considered. This method consists of estimating the value of the analysed parameter when the cell quantity tends to infinite from a minimum of three meshes. Therefore, a coarse mesh (16,809 cells), a medium mesh (25,665 cells), and a fine mesh (43,963 cells) were considered. As summarized in Table 2, the convergence condition (R), which should be between 0 and 1 to ensure a monotonic convergence, is fulfilled, and the estimated values (RE) of the evaluated parameters are close to the ones obtained with the fine mesh. Therefore, the mesh is suitable for these simulations. In addition, the results were compared with the experimental ones of Roshko et al. [22] for Re = 6383, showing fairly similar values.

Regarding the fluid, incompressible turbulent unsteady air is considered. The density (ρ) of the fluid is equal to 1.18415 kg/m^3, and its dynamic viscosity (μ) is equal to 1.85508·10^{-5} Pa·s. These values are assumed to be constant. The velocity at the inlet (u_∞) is set at 5 m/s, which means that the Reynolds number (Re) ranges between 6380 and 12,760, depending on the geometry and according to Expression (1).

$$Re = \frac{u_\infty \cdot L \cdot \rho}{\mu} \tag{1}$$

where L is the projection of the geometry on the direction of the flow.

Table 1. Tested geometries.

Shape	Sketch	Orientation	Scale	Number of Data
Circle		-	a = 0.02 m, 0.022 m, . . . , 0.04 m	11
Equilateral triangle		0°, 3°, . . . , 177°	a = 0.02 m	60
Square		0°, 3°, . . . , 87°	a = 0.02 m	30
Equilateral pentagon		0°, 3°, . . . , 69°	a = 0.02 m	24
Equilateral hexagon		0°, 3°, . . . , 57°	a = 0.02 m	20
Ellipse		0°, 3°, . . . , 177°	a = 0.02 m; b/a = 1.1, 1.2, . . . , 2	600
Rectangle		0°, 3°, . . . , 177°	a = 0.02 m; b/a = 1.1, 1.2, . . . , 2	600
Triangle		0°, 3°, . . . , 357°	a = 0.01 m; b/a = 1.5, 1.75; γ = 40°, 60°, 80°	720

For turbulence modelling, the RANS-based realizable k-ε two-layer [23] turbulence model is selected, since k-ε models are the most common ones to obtain mean flow characteristics for turbulent flow conditions. RANS turbulence models provide closure relations for the RANS equations that govern the transport of the mean flow quantities. To obtain these equations, each flow variable is divided into a mean value and its fluctuating compo-

nent, and then, the mean values are inserted into the Navier-Stokes equations, obtaining the mean mass and momentum transport Equations (2) and (3).

$$\nabla \cdot \overline{u} = 0 \tag{2}$$

$$\frac{\partial}{\partial t}(\rho \overline{u}) + \nabla \cdot (\rho \overline{u} \otimes \overline{u}) = -\nabla \cdot \overline{p} I + \nabla \cdot (T + T_{RANS}) + f_b \tag{3}$$

where $\overline{u}$ and $\overline{p}$ are the mean velocity and pressure, respectively; I is the identity tensor; T is the viscous stress tensor; and f_b is the body force.

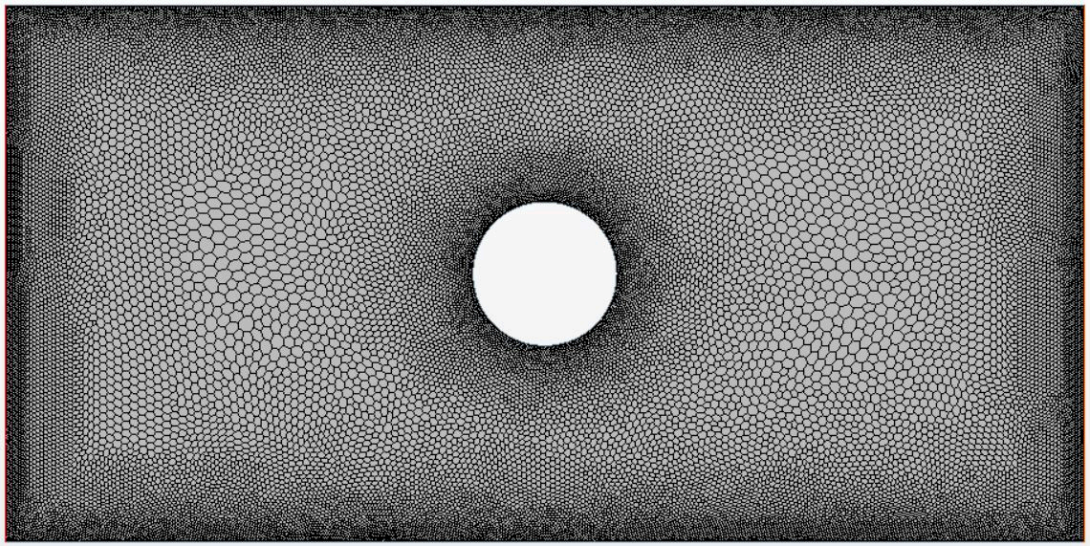


Figure 2. Mesh for a circle-shaped geometry.

Table 2. Mesh verification and comparison with experimental data for the case of a cylinder with a = 0.02 m.

Mesh Resolution			Richardson Extrapolation			Experimental
Coarse	**Medium**	**Fine**	**RE**	***p***	**R**	
0.796	0.835	0.858	0.907	0.681	0.566	0.91

Depending on the modelling of the stress tensor, there are different RANS model categories. The *k*-ε model corresponds to the eddy viscosity models, which are based on the analogy between the molecular gradient-diffusion process and turbulent motion. This kind of models uses the turbulent dynamic eddy viscosity (μ_t) to model the stress tensor as a function of mean flow quantities. In the present case, T_{RANS} is modeled by means of the Boussinesq approximation (4).

$$T_{RANS} = 2\mu_t S - \frac{2}{3}(\mu_t \nabla \cdot \overline{u}) I \tag{4}$$

where S is the mean strain rate tensor defined by Equation (5).

$$S = \frac{1}{2}\left(\nabla \cdot \overline{u} + \nabla \cdot \overline{u}^T\right) \tag{5}$$

The RANS-based *k*-ε model is a two-equation model which consists of the model transportation equation for turbulent kinetic energy (k) (5), the model transportation equation for the dissipation rate (ε) (6) which is empirical, and the turbulent viscosity (μ_t) specification (7).

$$\frac{\partial}{\partial t}(\rho k) + \nabla \cdot (\rho k \overline{u}) = \nabla \cdot \left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\nabla k\right] + P_k - \rho\varepsilon \tag{6}$$

$$\frac{\partial}{\partial t}(\rho\varepsilon) + \nabla\cdot(\rho\varepsilon\overline{u}) = \nabla\cdot\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\nabla\varepsilon\right] + \frac{\varepsilon}{k}C_{\varepsilon 1}P_\varepsilon - \rho C_{\varepsilon 2}\frac{k}{k+\sqrt{v\varepsilon}}\left(\frac{\varepsilon^2}{k}\right) \tag{7}$$

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \tag{8}$$

where σ_k, σ_ε, $C_{\varepsilon 1}$, $C_{\varepsilon 2}$, and C_μ are model coefficients; P_k and P_ε are production terms defined by (9) and (10), respectively; and v is the kinematic viscosity ($v = \mu/\rho$).

$$P_k = G_k + G_b + \gamma_M \tag{9}$$

$$P_\varepsilon = G_k + C_{\varepsilon 3}G_b \tag{10}$$

where $C_{\varepsilon 3}$ is a model coefficient, whose value is 1 if $G_b \geq 1$ and 0 if $G_b < 0$; G_k represents the turbulent production given by Equation (11); G_b represents the buoyancy production given by Equation (12); and γ_M represents the compressibility modification given by Equation (13).

$$G_k = \mu S^2 - \frac{2}{3}\rho k\nabla\cdot\overline{u} - \frac{2}{3}\mu_t(\nabla\cdot\overline{u})^2 \tag{11}$$

$$G_b = -\frac{1}{\rho}\cdot\frac{\partial\rho}{\partial\overline{T}}\cdot\frac{\mu_t}{Pr_t}(\nabla\overline{T}\cdot g) \tag{12}$$

$$\gamma_M = \frac{C_M k\varepsilon}{c^2} \tag{13}$$

where Pr_t is the turbulent Prandtl number; $\overline{T}$ is the mean temperature; g is the gravitational vector; C_M is a model coefficient equal to 2; and c is the speed of sound.

Among the various forms of the *k*-ε model that are available, the realizable *k*-ε two-layer model is considered in the present study. This model combines the realizable *k*-ε model, which satisfies certain mathematical constraints on the normal stresses consistent with the physics of turbulence, and the two-layer approach, which allows the *k*-ε model to be applied in the viscous-affected layer. With the realizable *k*-ε two-layer model, the coefficients are identical to the ones of these two models separately, but the model gains the added flexibility of an all-y^+ wall treatment. For the selected turbulence model and the studied cases, the model coefficients are the following ones: σ_k is equal to 1; σ_ε is equal to 1.2; $C_{\varepsilon 1}$ is equal to 1.44; $C_{\varepsilon 2}$ is equal to 1.9; and C_μ is equal to 0.09.

Regarding the wall treatment, as mentioned before, the realizable *k*-ε two-layer model uses an all-y^+ wall treatment. This wall treatment emulates the low-y^+ wall treatment for fine meshes (near the boundaries), which resolves the viscous sublayer and needs little or no modelling to predict the flow across the wall boundary; the high-y^+ wall treatment for coarse meshes (far from the boundaries), which, instead of resolving the viscous sublayer, obtains the boundary conditions for the continuum equations.

For data generation, as non-stationary turbulence models are selected, the average values of the velocity, pressure, and vorticity fields are extracted. To obtain these average fields, 2 s of simulation are considered, once the flow is fully developed. The values of the fields are interpolated in order to fit the data into a 128 × 256 grid. Then, the procedure of Kashefi et al. [14] for data generation is followed.

First, the values are normalized following Expressions (14)–(17), to get dimensionless values.

$$u_x^* = \frac{u_x}{u_\infty} \tag{14}$$

$$u_y^* = \frac{u_y}{u_\infty} \tag{15}$$

$$p^* = \frac{p}{\rho\cdot u_\infty^2} \tag{16}$$

$$\omega^* = \frac{\omega\cdot a}{u_\infty} \tag{17}$$

where u_x^*, u_y^*, p^*, and ω^* are the dimensionless variables.

Finally, once the variables are dimensionless, they are scaled in a range of (0, 1), following Expression (18).

$$\Phi' = \frac{\Phi - \min(\Phi)}{\max(\Phi) - \min(\Phi)} \tag{18}$$

where Φ is replaced by each set of u_x^*, u_y^*, p^*, and ω^*.

2.2. Convolutional Neural Network

2.2.1. Domain Representation

In this study, the same layers used by Ribeiro et al. [13] are implemented to represent the domain. Therefore, numerical domain is represented by three different layers: the Flow Region Channel (FRC), the Signed Distance Function (SDF) of the geometry, and the SDF of the walls.

The FRC layer contains information about the boundary conditions of the domain. This layer consists of giving a number to each cell of the grid depending on the boundary condition assigned to that cell. In this case, the geometry is represented by a 0, the fluid by a 1, no-slip walls by a 2, the inlet by a 3, the outlet by a 4, and the outline of the geometry by a 5. A detailed example of a FRC layer is shown Figure 3.

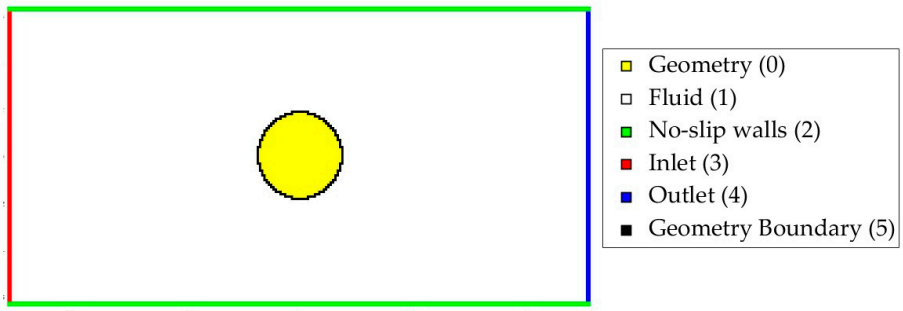
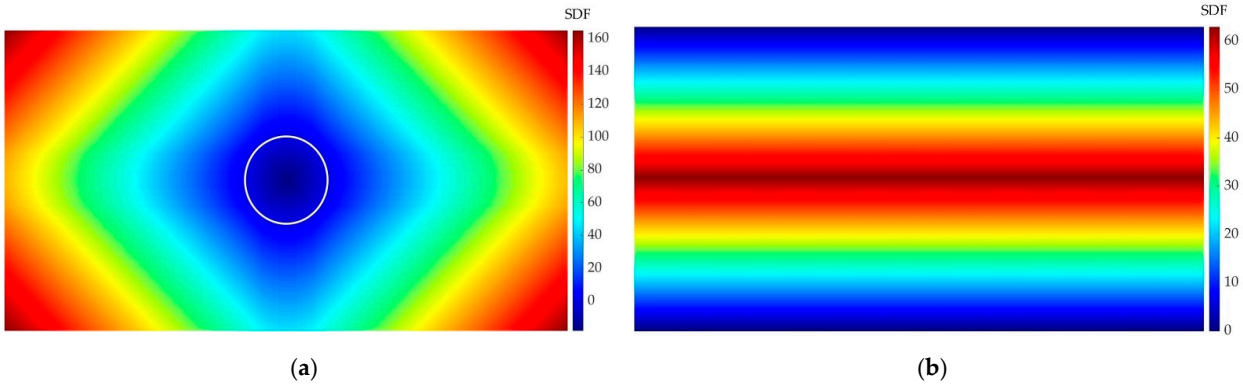


Figure 3. Flow Region Channel representation (not to scale).

The SDF layer represents the minimum distance between each cell and the outline of a specified contour. This function was proposed by Guo et al. [8], and as demonstrated in that study, it provides significantly smaller errors than the typical binary representation. In this study two different SDF layers are used, one for the geometry, shown in Figure 4a, and another one for the no-slip walls of the channel, shown in Figure 4b.



(**a**) (**b**)

Figure 4. SDF layer examples. (**a**) SDF of the geometry (the outline of the geometry is represented in white); (**b**) SDF of the channel walls.

To create this layer, firstly the zero level set (Z) is created (19). This is the level where the analysed contour is located, and therefore, SDF is equal to zero.

$$Z = \left\{ (X, Y) \in R^2 / \; SDF(x, y) = 0 \right\} \tag{19}$$

Then, the sign of SDF is defined. $SDF(x, y) = 0$ if (x, y) is on the geometry contour; $SDF(x, y) < 0$ if (x, y) is inside the geometry; and $SDF(x, y) > 0$ if (x, y) is outside the geometry.

Finally, the value of each cell is calculated following Expression (20).

$$SDF(x, y) = \min_{(X,Y) \in Z} |(x, y) - (X, Y)| \cdot sign \tag{20}$$

After generating all the input layers, they are scaled in a range of (0, 1), following the previously-mentioned Expression (18).

2.2.2. Neural Network Architecture

In the present paper, a CNN based on the previous works from Ribeiro et al. [13] and Thuerey et al. [18] is proposed. For this network, an U-Net architecture [24] is considered, which is a special case of an encoder-decoder network. In this case, the net consists of four encoder/decoder blocks. Each encoder block contains two convolutional layers. The first convolutional layer is followed by a ReLU (Rectifier Linear Unit) layer, and the second one by a ReLU layer and a Max Pooling layer. In the first two encoding blocks, the kernel size is equal to 5, and strided convolutions are performed on the first layer of the block, aiming to reduce the data size for the training step. In contrast, the kernel size of the las two encoding blocks is equal to 3. After each block, the number of filters is doubled. The decoding phase performs the reverse process. Encoder and decoder blocks are connected to each other by concatenation layers. A schematic view of the used CNN is provided in Figure 5. MATLAB 2021a [25] commercial code with its Deep Learning Toolbox [26] was used for designing and training the network.

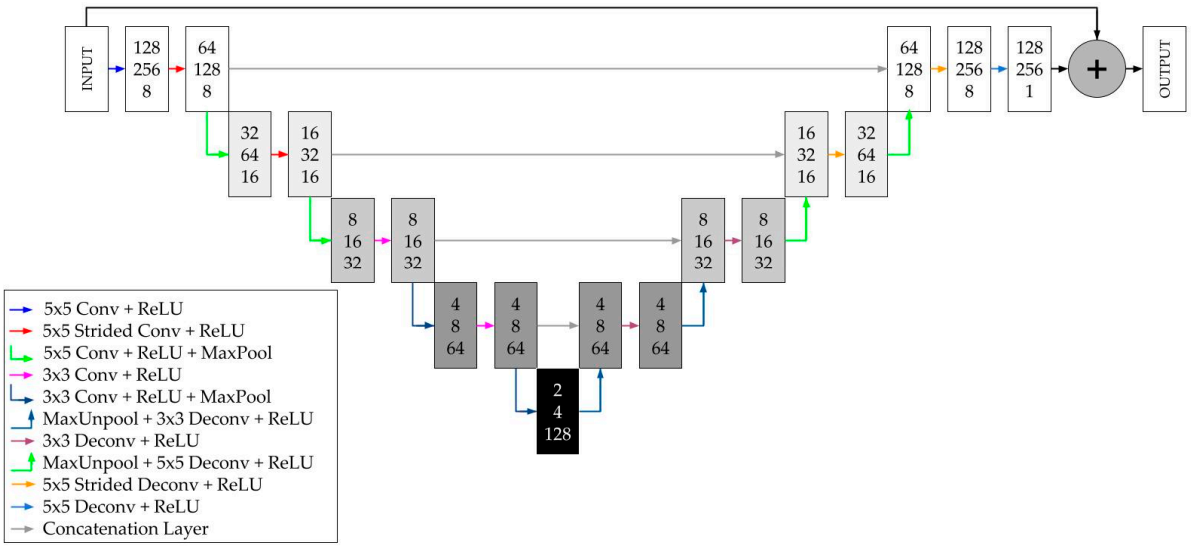


Figure 5. CNN architecture.

With regards to the network training process, the Adam [27] optimizer was used, with a learning rate of 0.001, a batch size of 64, and a weight decay of 0.005. The data were split into 60% training, 30% validation, and 10% test, and the validation was performed after each epoch. Three different configurations have been considered for the selection of the most appropriate data-splitting ratio: 60% training, 30% validation, and 10% test (the selected ratio); 70/20/10%; and 80/10/10%. Among these configurations, the selected

one is the one that provides the best predictions of all the analysed magnitudes. Among these magnitudes, pressure is the most sensitive to the data-split, and vorticity the least sensitive, the differences between configurations being almost negligible for this magnitude. Considerable differences also appear in the velocity fields.

In order to validate the used net, the training Root-Mean Square Error (RMSE) curves obtained with the network used in the present study are compared with the ones obtained with the net of Ribeiro et al. [13], which was designed for laminar flow prediction on a channel. As this network was originally designed to predict velocity and pressure fields, only the training curves of these magnitudes are considered for this comparison. These curves can be shown in Figure 6.

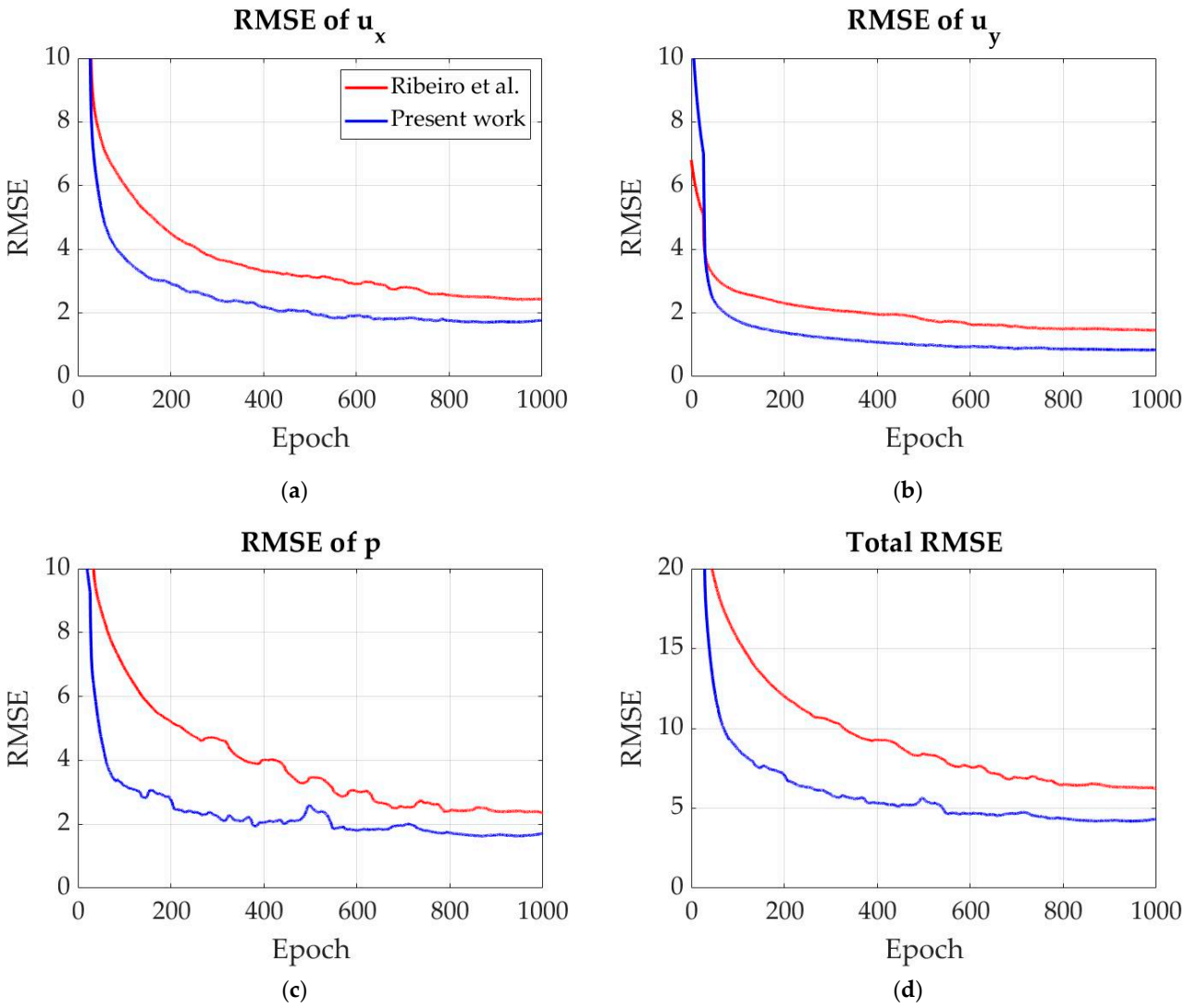


Figure 6. Comparison of the obtained training error curves with the ones obtained with the net of Ribeiro et al. [13] for 1000 epochs. (**a**) RMSE or u_x; (**b**) RMSE of u_y; (**c**) RMSE of p; (**d**) Total RMSE.

The curves show that the proposed network outperforms the baseline network for turbulent flow predictions. All the curves show a broadly similar trend. At the beginning of the training, up to approximately epoch 50, the error of the proposed network decreases significantly more than the error of the baseline network, except in the case of u_y. Thereafter, the difference between the two networks narrows, but when the results stabilize, the error of the proposed network is still lower. At the end of training, the proposed network has an error reduction of 28% in the case of u_x, 42% in u_y, and 30% in p.

2.2.3. Neural Network Configurations

In the present study, four different network structures are proposed in order to determine which provides the best predictions of the turbulent flow velocity fields. All structures have the domain characteristics as input and the velocity fields as output, but three of them have an additional intermediate stage to calculate the pressure and vorticity fields.

The basic structure (Figure 7a) is the most popular for predicting flow characteristics. This structure directly predicts the velocity fields. The pressure-based structure (Figure 7b) predicts the pressure field, and then the velocity fields considering the pressure field. The vorticity-based structure (Figure 7c) and pressure- and vorticity-based structure (Figure 7d) are equal to the pressure-based one, but these networks predict the vorticity field and both pressure and vorticity fields in the intermediate stage, respectively.

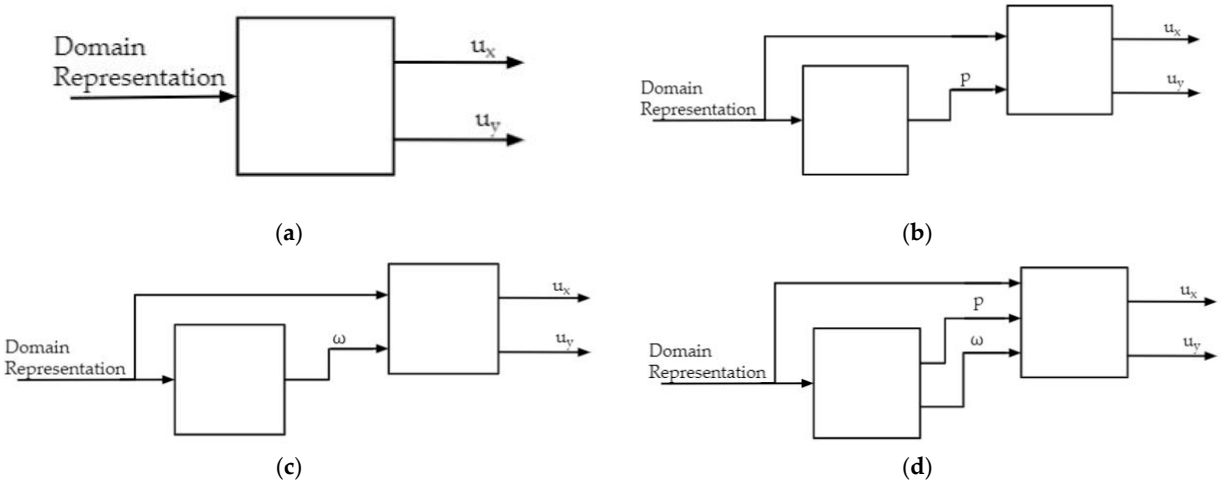


Figure 7. Studied ANN structures. (**a**) Basic structure; (**b**) Pressure-based structure; (**c**) Vorticity-based structure; (**d**) Pressure- and vorticity-based structure.

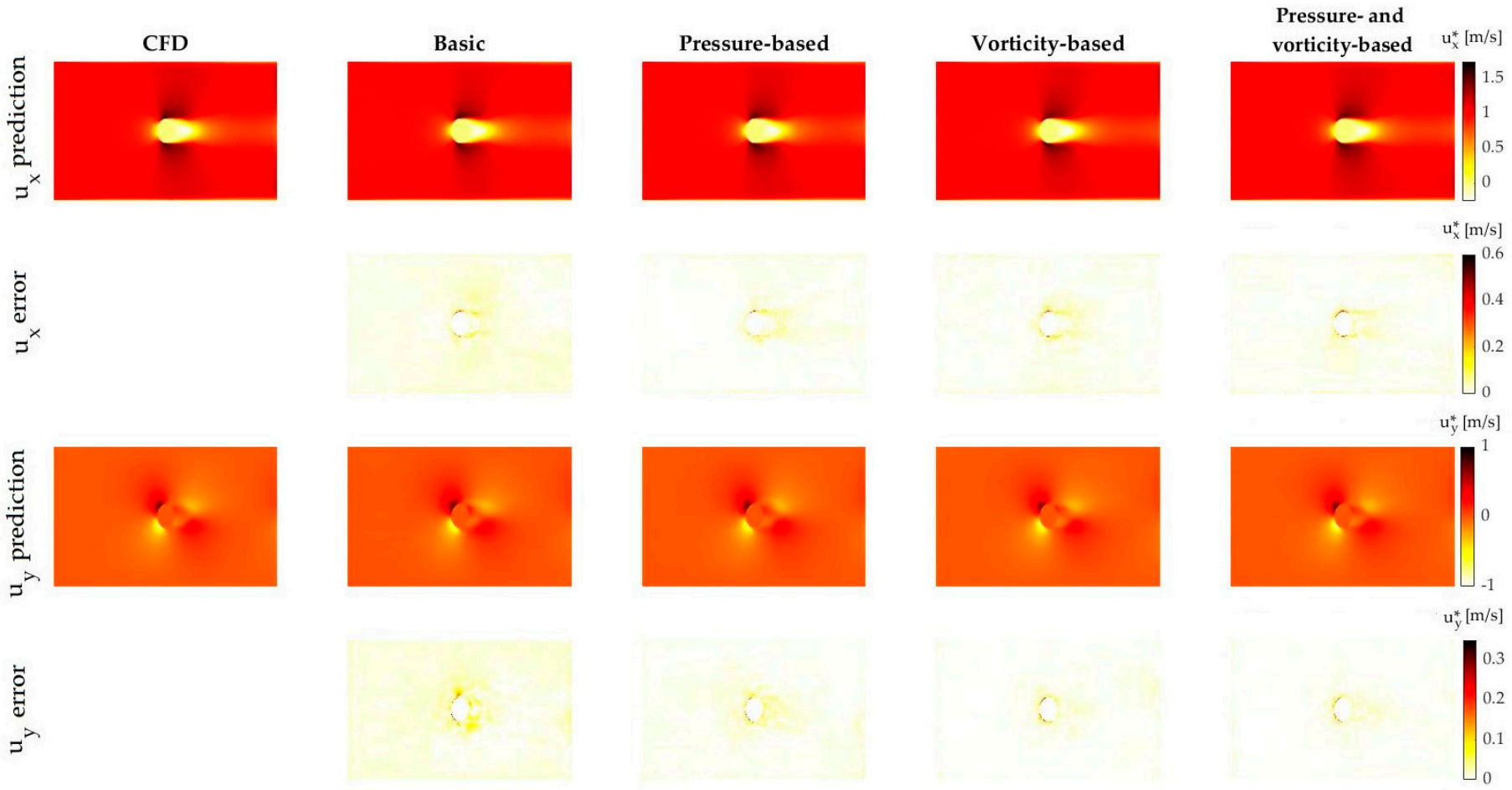
These magnitudes are selected for the intermediate stage as they are directly related to velocity. Regarding pressure, according to Bernoulli's principle, a decrease in the pressure occurs simultaneously with an increase in the velocity, and vice versa. Concerning vorticity, this magnitude determines the local rotation of the fluid, thus relating the velocity components.

3. Results and Discussion

Aiming to determine which of the previously mentioned ANN configurations provides the best predictions of turbulent velocity fields, a qualitative and quantitative comparison of the results obtained with all the structures is conducted. For these two studies, the results obtained with a test-set of 207 geometries (10% of all the samples) are considered.

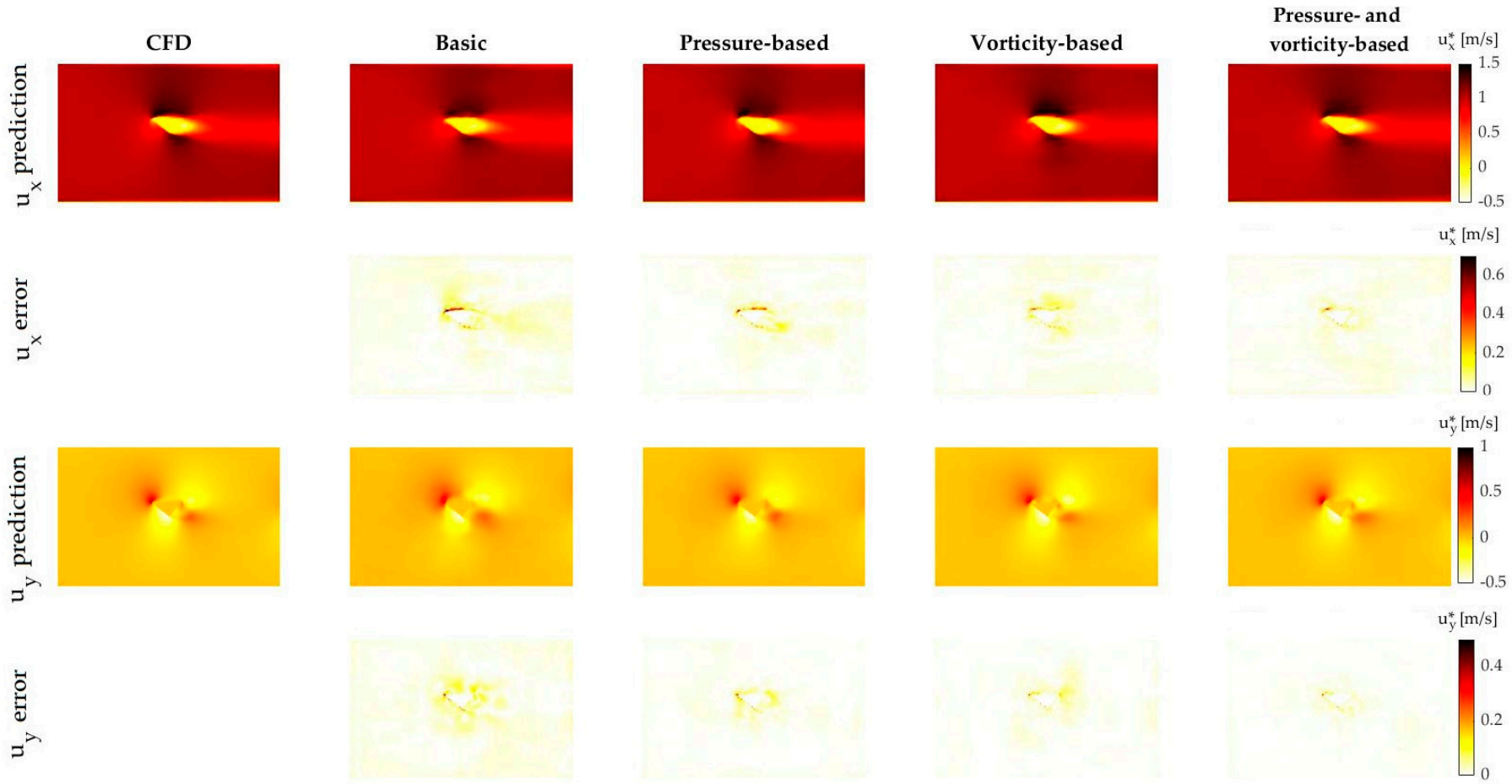
3.1. Qualitative Study

The qualitative study is performed by comparing the predictions of the studied neural network structures with the results obtained by CFD simulations. To conduct this comparison, three different geometries, with different characteristics, are selected. The first geometry (Figure 8a) is an ellipse, and it is considered an easy-to-predict geometry because of its symmetry, its low Reynolds number and the fact of not having sharp corners. The second geometry (Figure 8b) is a triangle, and it is considered an aerodynamic geometry due to its low angle of attack and the small surface area on which the fluid directly impinges. The third geometry (Figure 8c) is also a triangle, and it is considered a non-aerodynamic geometry because of its high angle of attack and the big area where the fluid directly impacts.



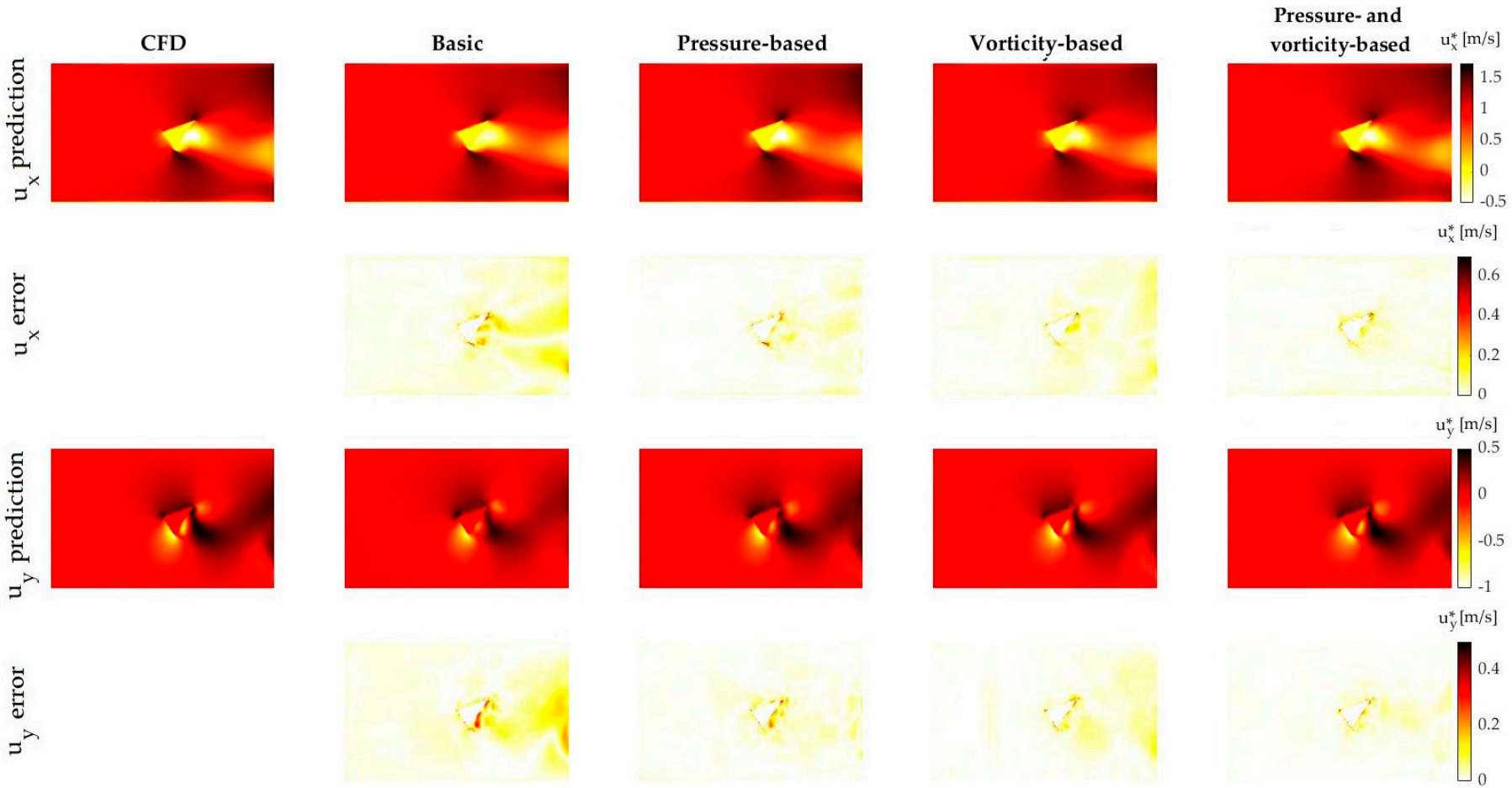
(**a**)

Figure 8. *Cont.*



(**b**)

Figure 8. *Cont*.



(**c**)

Figure 8. Qualitative comparison of the results obtained with the studied structures. (**a**) Easy-to-predict geometry; (**b**) Aerodynamic geometry; (**c**) Non-aerodynamic geometry.

The results show that, although some errors appear in the contour and in the wake behind the geometry, all the structures are able to predict the velocity fields of the easy-to-predict geometry and the aerodynamic geometry fairly precisely. For these two geometries, the basic structure is the one which provides the worst results, being the structure which shows the highest errors. Between the other three structures, no clear differences are visible, but the pressure- and vorticity-based structure seems to be the most accurate structure with very low errors across the predicted fields.

The predictions of the non-aerodynamic geometry show the larger differences between structures. In this case, the basic structure is clearly outperformed by the other ones, showing poor predictions on the boundary of the geometry and, most markedly, on the wake behind the geometry. This means that giving additional information about the flow to the network eases predictions when the flow behaviour is uncertain. The networks which include the prediction of the pressure show the best results on the wake behind the geometry, and the ones which include the prediction of the vorticity show the best results near the geometry. This is attributed to the fact that changes in the pressure field start to appear at a considerable distance from the geometry, but the largest changes in the vorticity field occur very close to the geometry, at its contour.

3.2. Quantitative Study

The quantitative study is focused on studying the error of the test-set. For a general overview of the error in the whole test-set, the mean error obtained with all the structures is analysed. Table 3 summarizes the mean error of the predicted velocity components.

Table 3. Mean absolute error of the velocity fields predicted by the tested structures.

Structure	Mean Absolute u_x Error (m/s)	Mean Absolute u_y Error (m/s)
Basic	0.1145	0.0851
Pressure-based	0.0619	0.0466
Vorticity-based	0.067	0.0331
Pressure- and vorticity-based	0.0636	0.0302

The mean error shows that all the proposed structures outperform the basic one. The proposed networks diminish the mean absolute u_x error between 41.5% and 45.9%, the pressure-based structure being the most accurate structure. Regarding the mean absolute u_y error, the proposed structures reduce the error between 45.2% and 64.5%, the pressure- and vorticity-based structure being the one which provides the best predictions.

In order to obtain more detailed information about the error distribution in the test-set, histograms with the absolute error of the velocity fields in both directions are made. These histograms can be shown in Figure 9.

The histograms of the absolute error confirm the results shown with the mean absolute errors. Although there is not a significant difference in the maximum u_x error obtained with the tested structures, the predictions made with the basic structure have the highest errors in almost all ranges above 0.05 m/s. As for the rest of the structures, although they show quite similar error distributions, the pressure- and vorticity-based structure is the structure with the least errors up to 2 m/s, but above this value, this structure produces more errors than the other two structures. On the other hand, the pressure-based structure has more errors below 2 m/s, but fewer above this value. For that reason, the mean error of the pressure-based structure is the lower one.

The predictions of all structures show a very similar trend of u_y error distribution, with the basic structure having the largest errors and the pressure- and vorticity-based structure having the smallest ones.

3.3. Performance Analysis

The main goal of calculating flow characteristics through neural networks is to reduce the high computational cost of CFD simulations. For this reason, the time required by each structure to make the predictions is considered a parameter of great relevance. Table 4

shows the time required to obtain the predictions with each studied method using a single core of an Intel Xeon 5420 CPU.

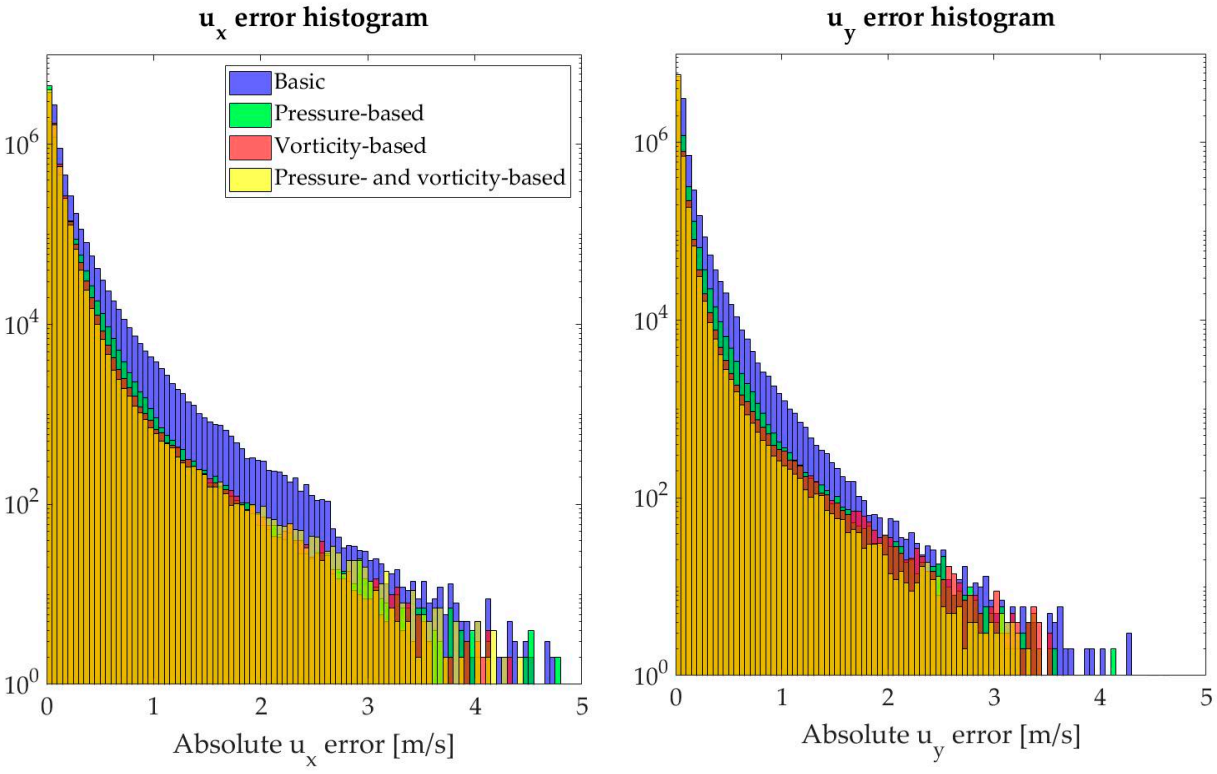


Figure 9. Absolute error distribution. (**a**) Absolute u_x error; (**b**) Absolute u_y error.

Table 4. Computational time required by each studied method to predict the velocity fields.

Method	Time (s)	Speedup
CFD	23,053.2	-
Basic	21.4	1077.25
Pressure-based	27.35	842.9
Vorticity-based	24.01	960.15
Pressure- and vorticity-based	28.53	808.03

The computational times required by each method show that predicting velocity fields using CNN, in comparison with CFD, entails a reduction of around four orders of magnitude in terms of computational time. As expected, the simplest CNN structure (the basic structure) is the fastest one, and the most complex (the pressure- and vorticity-based structure) is the slowest one. Nonetheless, the differences between structures are negligible in comparison with the time required by CFD.

4. Network Testing

In order to evaluate the ability of the neural network proposed in this study to make predictions under different conditions from the training ones, two different domains are considered, one with two geometries and another one with a different channel.

To make the predictions of the velocity fields on these domains, the best neural network structure is considered. For predicting the u_x velocity field, the pressure-based structure is considered, since it provides the minimum mean error and is the better one for predicting the wake behind the geometry, which is one of the most important parameters. To predict the u_y velocity field, the pressure- and vorticity-based structure is considered, since it has been demonstrated to be the most accurate for this field. A schematic view of the structure of the used ANN is provided in Figure 10.

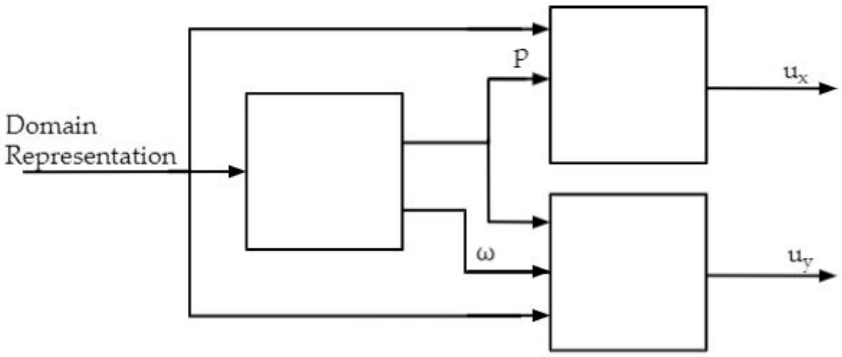
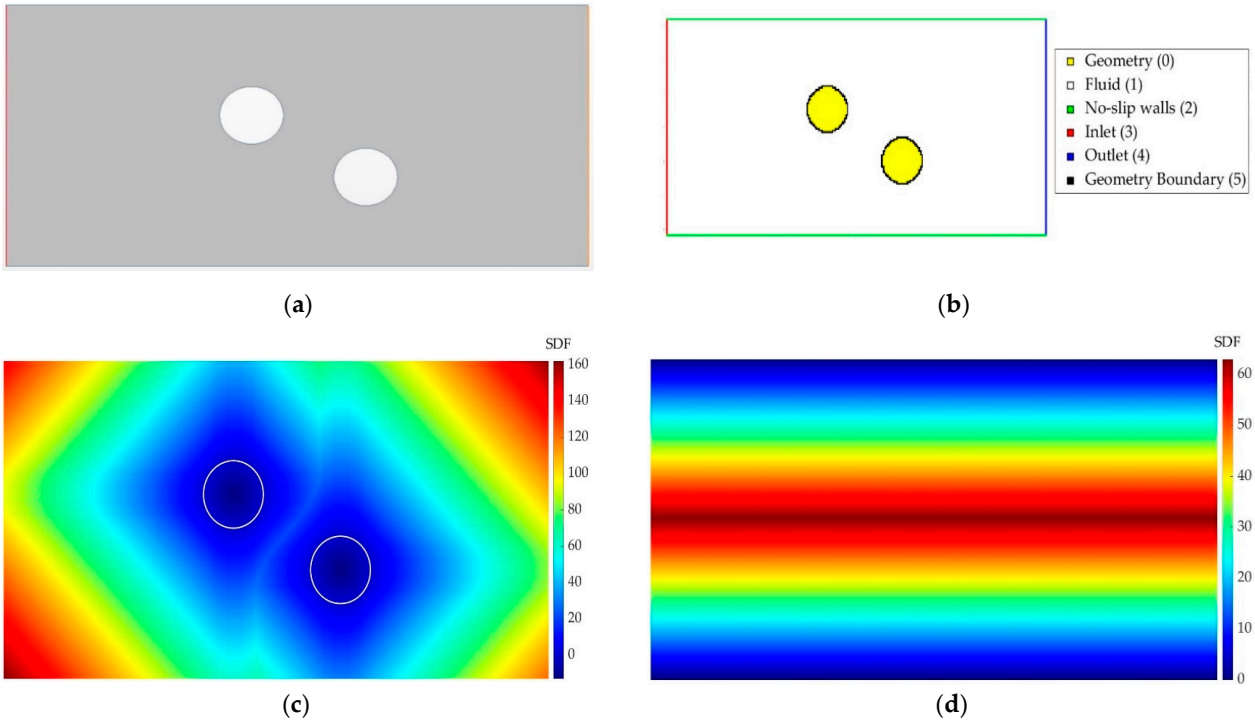


Figure 10. ANN structure to test the proposed network under different conditions.

4.1. Multiple Objects

The domain used in this case contains two geometries, particularly two circles. Therefore, the input layers for this case have some adaptations. The FRC layers contain two geometries and geometry boundaries, and the SDF layer of the geometry has two zero level sets. The SDF of the walls remains equal. A schematic view of the numerical domain and the input layers is provided in Figure 11. The predictions of the velocity fields performed by CFD and the proposed network are displayed in Figure 12.



(**a**) (**b**)

(**c**) (**d**)

Figure 11. Information about the used domain with multiple geometry. (**a**) Numerical domain; (**b**) FRC layer; (**c**) SDF layer of the geometries (the contour of the geometries is drawn in white); (**d**) SDF layer of the walls.

The results show that the proposed network is able to predict the velocity fields of a domain with multiple geometries in a fairly reliable way, with tolerable errors. These results show two problematic areas. The first problematic area is the contour of the geometries. Whereas with a single geometry, these errors were already visible; in this case they are slightly higher. The second problematic area is the area between geometries. Although the errors are not very high, the neural network does not adequately predict the interaction between geometries, since the area with the most errors is the region where the wake of the first geometry impacts the second geometry.

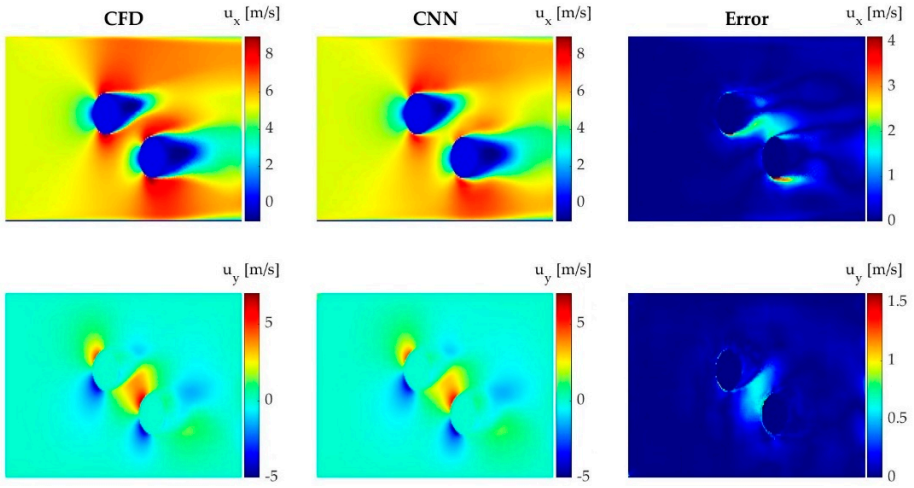


Figure 12. Comparison of the velocity fields of a domain with two geometries obtained with CFD and the proposed net.

4.2. Different Channel

This domain contains a single geometry, but the walls of the channel have a narrowing on their middle. Therefore, the input layers for this case also have to be adapted. In the FRC layer, although the outline of the walls remains with no-slip wall conditions, the inner part of the walls has been considered as geometry, since among the possible options it is considered to be the most appropriate one. The SDF layer of the geometry is equal to the training ones, but the SDF layer of the walls contains a zone of negative values. A schematic view of the numerical domain and the input layers is provided in Figure 13, and the predictions of the velocity fields performed by CFD and the proposed network are displayed in Figure 14.

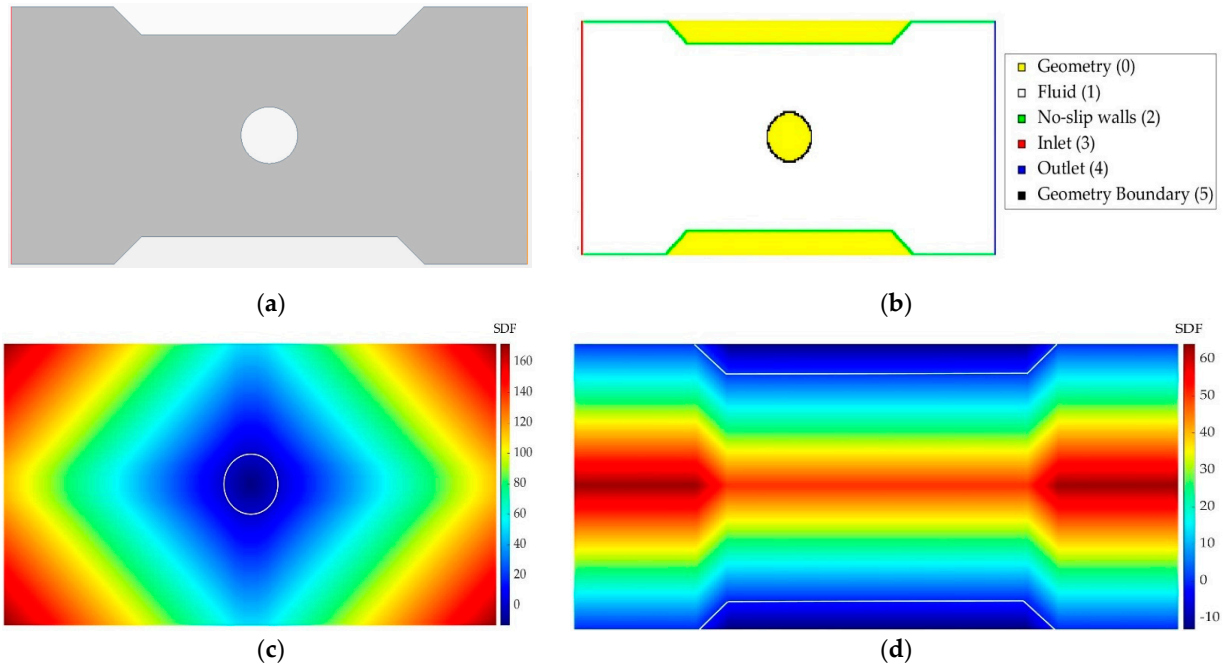


Figure 13. Information about the used domain with a different channel. (**a**) Numerical domain; (**b**) FRC layer; (**c**) SDF layer of the geometry (the contour of the geometry is drawn in white); (**d**) SDF layer of the walls (the contour of the walls is drawn in white).

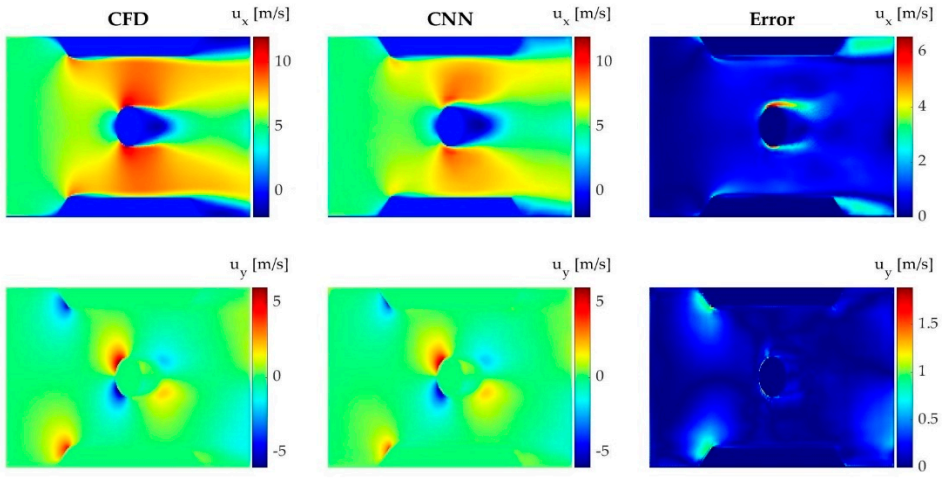


Figure 14. Comparison of the velocity fields of a domain with a different channel obtained with CFD and the proposed net.

There are three areas where considerable differences are observed. The first zone is the narrowing of the channel, where the neural network underpredicts the speed increase that occurs. The second, and most conflicting, area is the near-wall area. When the channel narrows, the network underpredicts the impact with the walls then (noticeable in u_y), and when the channel widens again, although a decrease in velocity is visible, the network is unable to predict the wake behind the walls (noticeable in u_x). The third region is the geometry contour. As with multiple geometries, when changing the channel, the errors in the geometry contour increase in comparison to the simple case.

5. Conclusions

In the present paper, a CNN for predicting different magnitudes of turbulent flows is proposed. With this CNN, alternative neural network structures for turbulent flow velocity field prediction are proposed. In contrast with the typical network structure, which directly calculated the velocity fields, the proposed structures perform a preliminary calculation of pressure and vorticity fields in order to obtain more information about the flow. Performing the predictions using the proposed networks instead of using CFD means a reduction of about four orders of magnitude in terms of computational time.

The results indicate that the proposed network structures outperform the basic structure, showing a decrease between 41.5% and 45.9% of mean absolute u_x error and a decrease between 45.2% and 64.5% of mean absolute u_y error. When the flow is simple, the results provided by the basic structure are correct, but the more uncertain the flow, the greater the differences between structures, with the structures with a preliminary calculation of pressure and vorticity fields being much more accurate than the basic one. These differences are more visible on the wake behind the geometry.

Finally, the best network structure is proposed considering the obtained results, and its ability to predict turbulent flow velocity fields in domains different from those used to train the network is evaluated. This network calculates the pressure and vorticity fields on the preliminary stage, and used the pressure field to calculate the horizontal component of the velocity field and both the pressure and vorticity fields to calculate the vertical component of the velocity field. To assess the network, two different domains are considered, one which has two geometries and another one which has a narrowing of the channel. The results show that the network is able to predict the velocity fields when the domain has two geometries, but when changing the channel, the errors are significant.

Therefore, this study demonstrates that obtaining fluid characteristics to predict the desired magnitude improves predictions substantially, and that this type of structure is applicable to different domains for the ones used for the training process.

Author Contributions: Conceptualization, K.P.-P.; methodology, K.P.-P. and A.U.-A.; software, E.Z. and A.Z.; validation, A.U.-A., E.Z. and U.F.-G.; formal analysis, K.P.-P. and A.U.-A.; investigation, K.P.-P. and A.U.-A.; resources, U.F.-G. and E.Z.; data curation, A.Z.; writing—original draft preparation, K.P.-P.; writing—review and editing, K.P.-P. and U.F.-G.; visualization, A.U.-A. and E.Z.; supervision, U.F.-G.; project administration, U.F.-G.; funding acquisition, U.F.-G. All authors have read and agreed to the published version of the manuscript.

Funding: The authors were supported by the government of the Basque Country through research grants ELKARTEK 21/10: BASQNET: Estudio de nuevas técnicas de inteligencia artificial basadas en Deep Learning dirigidas a la optimización de procesos industriales.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors are grateful for the support provided by SGIker of UPV/EHU. This research was developed under the frame of the Joint Research Laboratory on Offshore Renewable Energy (JRL-ORE).

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

	Definition
ANN	Artificial Neural Network
CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Network
DL	Deep Learning
DML	Deep Metric Learning
DNN	Deep Neural Network
DNS	Direct Numerical Simulation
FRC	Flow Region Channel
GAN	Generative Adversarial Network
LES	Large Eddy Simulation
RANS	Reynolds-Averaged Navier-Stokes
ReLU	Rectifier Linear Unit
RMSE	Root-Mean-Square Error
SDF	Signed Distance Function
*	Dimensionless value
‘	Value ranged between [0, 1]
–	Mean value
a	Characteristic length of the geometry
b	Characteristic length of the geometry
c	Speed of sound
$C_{\varepsilon 1}$	Model coefficient
$C_{\varepsilon 2}$	Model coefficient
$C_{\varepsilon 3}$	Model coefficient
C_{μ}	Model coefficient
C_M	Model coefficient
γ	Characteristic angle of the geometry
γ_M	Compressibility modification
ε	Dissipation rate
f_b	Resultant of body forces
g	Gravitational vector

G_b	Buoyancy production
G_k	Turbulent production
I	Identity tensor
k	Turbulent kinetic energy
L	Projection of the geometry on the direction of the flow
ρ	Fluid density
p	Pressure
p	Order of accuracy (in General Richardson Extrapolation)
P_k	Production term
P_ε	Production term
Pr_t	Turbulent Prandtl number
R	Convergence condition (in General Richardson Extrapolation)
Re	Reynolds number
RE	Estimated value (in General Richardson Extrapolation)
σ_k	Model coefficient
σ_ε	Model coefficient
T	Viscous stress tensor
$\overline{T}$	Mean temperature
μ	Fluid dynamic viscosity
μ_t	Turbulent viscosity
u_x	Horizontal velocity
u_y	Vertical velocity
ω	Vorticity
Z	Zero level set (in SDF)

References

1. Ray, D.; Hesthaven, J.S. An artificial neural network as a troubled-cell indicator. *J. Comput. Phys.* **2018**, *367*, 166–191. [CrossRef]
2. Liu, Y.; Lu, Y.; Wang, Y.; Sun, D.; Deng, L.; Wan, Y.; Wang, F. Key time steps selection for CFD data based on deep metric learning. *Comput. Fluids* **2019**, *195*, 104318. [CrossRef]
3. Bao, H.; Feng, J.; Dinh, N.; Zhang, H. Computationally efficient CFD prediction of bubbly flow using physics-guided deep learning. *Int. J. Multiph. Flow* **2020**, *131*, 103378. [CrossRef]
4. Hanna, B.N.; Dinh, N.T.; Youngblood, R.W.; Bolotnov, I.A. Coarse-Grid Computational Fluid Dynamic (CG-CFD) Error Prediction Using Machine Learning. *arXiv* **2017**, arXiv:171009105.
5. Yan, X.; Zhu, J.; Kuang, M.; Wang, X. Aerodynamic shape optimization using a novel optimizer based on machine learning techniques. *Aerosp. Sci. Technol.* **2019**, *86*, 826–835. [CrossRef]
6. Zhang, X.; Xie, F.; Ji, T.; Zhu, Z.; Zheng, Y. Multi-Fidelity deep neural network surrogate model for aerodynamic shape optimization. *Comput. Methods Appl. Mech. Eng.* **2021**, *373*, 113485. [CrossRef]
7. Tao, J.; Sun, G. Application of deep learning based multi-fidelity surrogate model to robust aerodynamic design optimization. *Aerosp. Sci. Technol.* **2019**, *92*, 722–737. [CrossRef]
8. Guo, X.; Li, W.; Iorio, F. Convolutional neural networks for steady flow approximation. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13 August 2016; pp. 481–490.
9. Ling, J.; Kurzawski, A.; Templeton, J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* **2016**, *807*, 155–166. [CrossRef]
10. Lee, S.; You, D. Prediction of laminar vortex shedding over a cylinder using deep learning. *arXiv* **2017**, arXiv:1712.07854.
11. Liu, Y.; Lu, Y.; Wang, Y.; Sun, D.; Deng, L.; Wang, F.; Lei, Y. A CNN-based shock detection method in flow visualization. *Comput. Fluids* **2019**, *184*, 1–9. [CrossRef]
12. Deng, L.; Wang, Y.; Liu, Y.; Wang, F.; Li, S.; Liu, J. A CNN-based vortex identification method. *J. Vis.* **2019**, *22*, 65–78. [CrossRef]
13. Ribeiro, M.D.; Rehman, A.; Ahmed, S.; Dengel, A. DeepCFD: Efficient steady-state laminar flow approximation with deep convolutional neural networks. *arXiv* **2020**, arXiv:2004.08826.
14. Kashefi, A.; Rempe, D.; Guibas, L.J. A Point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries. *arXiv* **2020**, arXiv:2010.09469.
15. Nowruzi, H.; Ghassemi, H.; Ghiasi, M. Performance predicting of 2D and 3D submerged hydrofoils using CFD and ANNs. *J. Mar. Sci. Technol.* **2017**, *22*, 710–733. [CrossRef]
16. Mohan, A.; Daniel, D.; Chertkov, M.; Livescu, D. Compressed convolutional LSTM: An efficient deep learning framework to model high fidelity 3D turbulence. *arXiv* **2019**, arXiv:1903.00033.
17. Fang, R.; Sondak, D.; Protopapas, P.; Succi, S. Deep learning for turbulent channel flow. *arXiv* **2018**, arXiv:1812.02241.
18. Thuerey, N.; Weißenow, K.; Prantl, L.; Hu, X. Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows. *AIAA J.* **2020**, *58*, 25–36. [CrossRef]

19. Sun, L.; Gao, H.; Pan, S.; Wang, J.-X. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Comput. Methods Appl. Mech. Eng.* **2020**, *361*, 112732. [CrossRef]
20. STAR-CCM+ V2019.1. Available online: https://www.plm.automation.siemens.com/ (accessed on 2 June 2020).
21. Richardson, L.F.; Gaunt, J.A. The deferred approach to the limit. *Philos. Trans. R. Soc. Lond. Ser. A Contain. Pap. Math. Phys. Character* **1927**, *226*, 299–361. [CrossRef]
22. Roshko, A. *Vortex Shedding from Circular Cylinder at Low Reynolds Number*; Cambridge University Press: Cambridge, UK, 1954.
23. Shih, T.-H.; Liou, W.W.; Shabbir, A.; Yang, Z.; Zhu, J. A new *k*-ε Eddy viscosity model for high Reynolds number turbulent flows: Model development and validation. *Computers Fluids* **1995**, *24*, 227–238. [CrossRef]
24. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Munich, Germany, 5–9 October 2015; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer: Cham, Switzerland, 2015; pp. 234–241.
25. MATLAB. Available online: https://es.mathworks.com/products/matlab.html (accessed on 9 June 2021).
26. Deep Learning Toolbox. Available online: https://es.mathworks.com/products/deep-learning.html (accessed on 3 July 2021).
27. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980.