MACHINE LEARNING BASED DRAG COEFFICIENT PREDICTION FOR TWO-DIMENSIONAL GEOMETRIES

BY

M A S MANURANGA

SUBMITTED TO THEDEPARTMENT OF MECHANICAL ENGINEERING IN THE PARITAL

FULFILLMENT OF THE REQUIREMENTS FOR

DEGREE OF

BACHELOR OF SCIENCE IN MECHANICAL ENGINEERING AT THE

UNIVERISTY OF SRI JAYEWARDENEPURA

JULY 2023

UNIVERSITY OF SRI JAYEWARDENEPURA

Faculty of Engineering

Module: ME4404

Thesis on Individual research project

Machine Learning Based Drag Coefficient Prediction for Any Two-Dimensional Geometry By

M A S Manuranga

ABSTRACT

With many potentials in the field of computational fluid dynamics, machine learning is rapidly rising to the top of the list of technologies for scientific computing. In present accurate prediction of aerodynamic properties is a key role for the design of applications that involve fluid flows. It is essential for the aerospace industry. In this paper it investigates the working of the convolutional neural network to predict the drag coefficient in laminar, low Reynolds number flow for generated two dimensional geometries. After the training aim is to optimize the efficiency of the trained convolutional neural network and increase the accuracy, decrease the validation loss of data set. Here we are focusing on the random two-dimensional shape drag coefficient. To find the drag capabilities from using machine learning we use convolutional neural network for laminar low Reynolds number flows which are associated with random two-dimensional shapes. Here our main goal is to train the machine learning model with convolutional neural network and evaluating the results with real life shapes.

Keywords—Drag coefficient, convolutional neural network, machine learning, Neural network.

Thesis Supervisor: Dr .Dulini Yasara Mudunkotuwa

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

ACKNOWLEDGEMENT

This thesis was prepared related to the ME4404 individual research project which is based on machine learning based drag coefficient prediction for two-dimensional geometry. The research about investigating the drag coefficient prediction of two-dimensional geometries using neural network with more accurate way.

I would like to express my deepest and sincere gratitude to individual research supervisor Dr. Dulini Yasara Mudunkotuwa , Head of the Department of Mechanical engineering , Senoir lecturer Department of Mechanical engineering University of Sri Jayewardenepura , for providing me with valuable guidance and support throughout the research project.

Also, I would like to acknowledge the co-supervisor Mr. Amila Karunanayake for guiding me as an instructor throughout this module in the Department of Mechanical Engineering University of Sri Jayewardenepura. Finally, I would like to thank everyone else who has been supported me throughout this individual research project.

M A S MANURANGA 18/ENG/064

Department of Mechanical Engineering,

Faculty of Engineering,

University of Sri Jayewardenepura,

Sri Lanka.

TABLE OF CONTENTS

Table of Contents

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Computational fluid dynamics is one of the emerging areas when considered about fluid dynamics. Computational fluid dynamics involves the use of algorithms and numerical methods to solve and analyze problems related to fluid flow. CFD also involves the use of computer software to simulate the behavior of fluids such as gases and liquids in various conditions and environments. Not only that CFD can be used to model and analyze a wide range of applications, including aerodynamics, automobile, pipe flows, hydrodynamics, combustion, heat transfer models and chemical reactions. For the behavior of fluids in complex geometries and environments also can be study for using Computational fluid dynamics because those studies are more difficult to study in an experimental environment. The basic steps in a CFD simulation include creating a computational mesh, specifying the fluid properties and boundary conditions, and solving the equations that describe the fluid flow and analyzing the results which gives us more accurate results in a simulation environment. CFD simulations can provide detailed insights into the behavior of fluids, and it can help engineers and scientists to design and optimize systems which are related with the fluids.

In present time there has been huge development in the use of neural networks and machine learning techniques in fluid dynamics and the field of computational fluid dynamics. These techniques have been applied in many ways which are replacement for certain steps in the resolution process such as pressure projection of closure terms in RANS (Reynolds Averages Navier Stocks) and LES (Large Eddy Simulations) computations.

One of the main advantages of using neural networks in computational fluid dynamics is that help to overcome limitations and the boundaries of traditional models by providing more accurate and generalized solutions. For example, in the case of physics informed deep learning which is a type of neural network is use two networks simultaneously to predict the PDE solutions and incorporate constraints from the original PDE (partial differential equation). This method can lead to better and high performance and more accurate solutions in computational fluid dynamics. Another application of neural network in CFD is as a surrogate model where a trained neural network can be used to predict flow properties and profiles of figures of merit as drag of lift. This is particularly useful and important in optimization problems or real-time decision processes where quick and accurate predictions are required.

Despite the recent hype around the use of neural network in CFD, there are still many challenges and limitations to be addressed. One of the main challenges is the need for substantial amounts of training data, as well as the development of appropriate architectures and training strategies. Moreover, the lack of interpretability of NN can make it difficult to understand the physical processes that are being modeled, and the potential impact of errors or uncertainties in the input data. When considering the drag and lift coefficient predictions using neural networks is a one of the emerging areas that uses, machine learning techniques in aerospace industry. From using large data airfoil profiles using the parameters of Reynolds number, Mach number, angle of attack it can train a machine learning model to predict better and accurate results from using less time than the computational operation. In addition, while the use of NN in CFD holds great promise, it is important to continue to explore the possibilities and limitations of these methods and to develop strategies for overcoming the challenges involved in their implementation.

The growing of the neural networks in the computational fluid dynamics area enhances various aspects of CFD simulation and analysis. When considering the neural networks of this area it can be trained to simplify the complex fluid flow problems. Instead of solving complex equations directly machine learning based neural networks can learn. The mapping between inputs and outputs of a particular problem. This advantage allows faster evaluation of the flow behavior without the use of computational simulations. When considering the turbulent flows, it is challenging to simulate in an accurate way because of its dynamic nature. When using neural networks that can develop data driven models that can execute the turbulent behavior in a more efficient way. It can also help to reduce the computational cost also. Not only that machine learning based neural networks also can be used to optimized flow control strategies and by connecting neural networks with optimized algorithms and it is easy to find the better shape and control variables that minimize the drag, and it helps to enhance the efficiency to reach the desired objectives.

For estimating the uncertainties of computational fluid dynamic simulations neural networks can be used. By self-learning from data that known uncertainties neural networks can do the predictions of flow properties which is an advantage of making decisions and can be perform the sensitive analysis also. It is especially important to identify that the application of neural networks to computational fluid dynamics is still an active key area of research and there are so many improvements that are happening to explore their capabilities and their limitations. The combination of computational fluid dynamics and neural networks is accelerating to improve accuracy and enabling the new pathways to fluid analysis and design improvements.

exciting area of research and has the potential to significantly improve the accuracy and efficiency of CFD simulations in a wide range of applications. Predicting the drag coefficient of two-dimensional shapes is an important problem in fluid mechanics, as it can help optimize the design of various objects, such as aircraft, automobiles, and ships. Machine learning can be used to develop predictive models that can accurately estimate the drag coefficient of arbitrary two-dimensional shapes.

There are some major applications based on two-dimensional drag prediction in engineering and physics. Predicting generated drag forces on airfoils and wings is important in aircraft design. From considering the flow characteristics and properties around two dimensional airfoils experts can understand the drag coefficient and optimize the shape for improved aerodynamic performance.

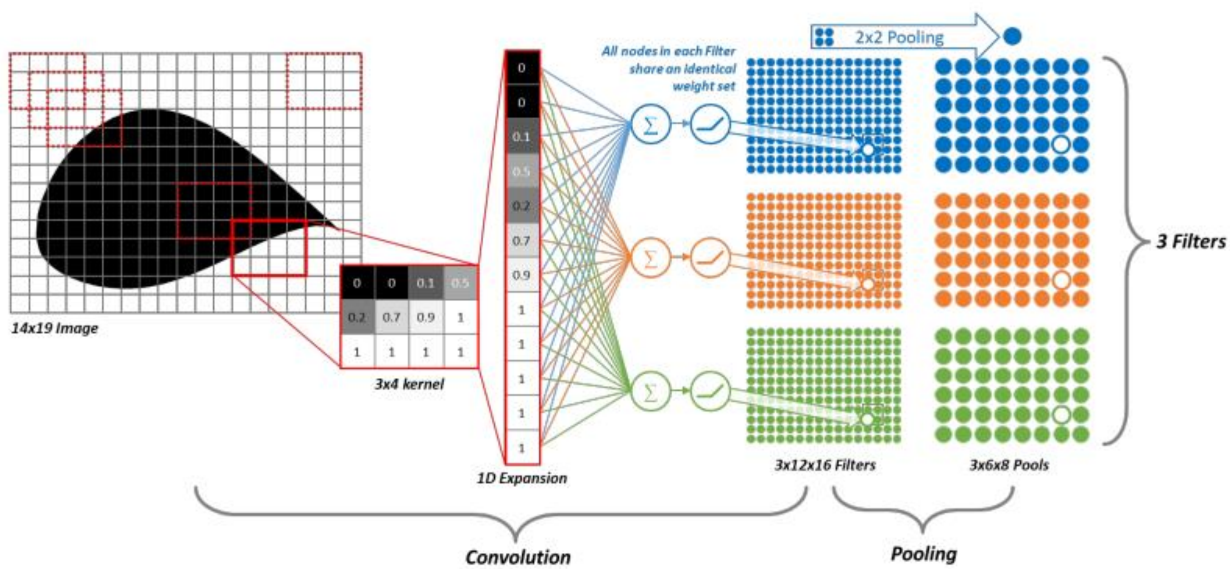

Figure 1: Two-dimensional airfoil image illustration of convolutional scheme

When considering the automobile industry, drag prediction is valuable in designing shapes of the cars to reduce the aerodynamic drag. To improve fuel efficiency engineers can optimize the shape of automobiles by using this kind of neural network drag prediction method. There are many fields that the drug will crucially affect. In sprots such as swimming and cycling reducing drag will affect to increase the performance. From considering flow around 2D shapes and models of sports equipment like swim kits, sky suits can help designers optimize their shape with minimizing the drag. Not only architects and civil engineers can utilize 2D drag predictions to give better performance of design buildings and structures to decrease the wind resistance. This will majorly affect the tall buildings and bridges where high drag forces can lead to structural failures of the buildings and bridges. Designing creative shapes and AI based design will be improved from using this kind of drag prediction method.

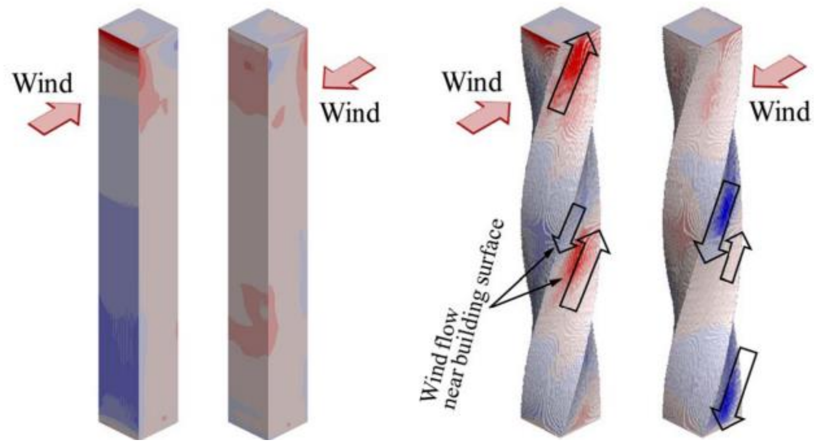

Figure 2: Wind flow behavior near buildings surface

LITERATURE REVIEW

The research papers use this neural network to develop computational simulations in a more accurate way. In here various convolutional architecture were used to study the aerodynamics. Lift and drag coefficient were predicted here by using neural network architectures. In this paper it mentioned that accurately predicting airfoil lift coefficient is most important for aircraft design and there were some limitations when using traditional methos. To overcome this problem, here authors propose using a convolutional model to predict airfoil lift coefficient based on airfoil shape and other necessary parameters such as angle of attack, Reynolds number, and the Mach number. They evaluate the performance of their convolutional neural network model on a dataset of more than 2000 airfoil shapes comparing it to traditional lift prediction methods such as XFOIL software and necessary relevant aerodynamic flow theory. The results of their study show that the CNN model outperforms traditional methods in terms of accuracy and computational power. Not only that, but the authors also note that the CNN model can be further developed by incorporating additional features such as surface roughness and turbulence efficiency.

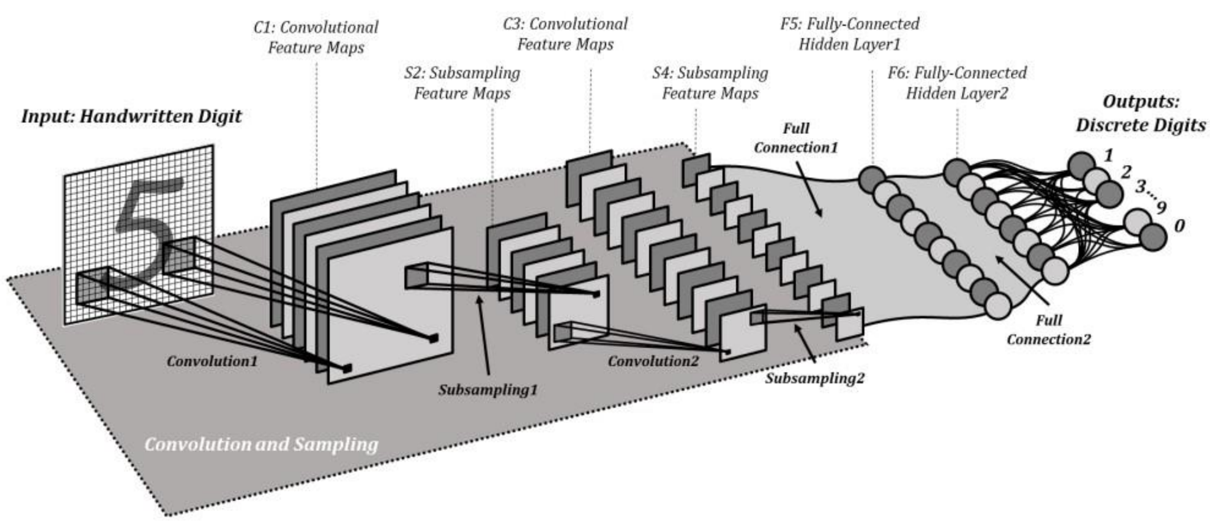

Figure 3: CNN for Handwritten digit recognition task

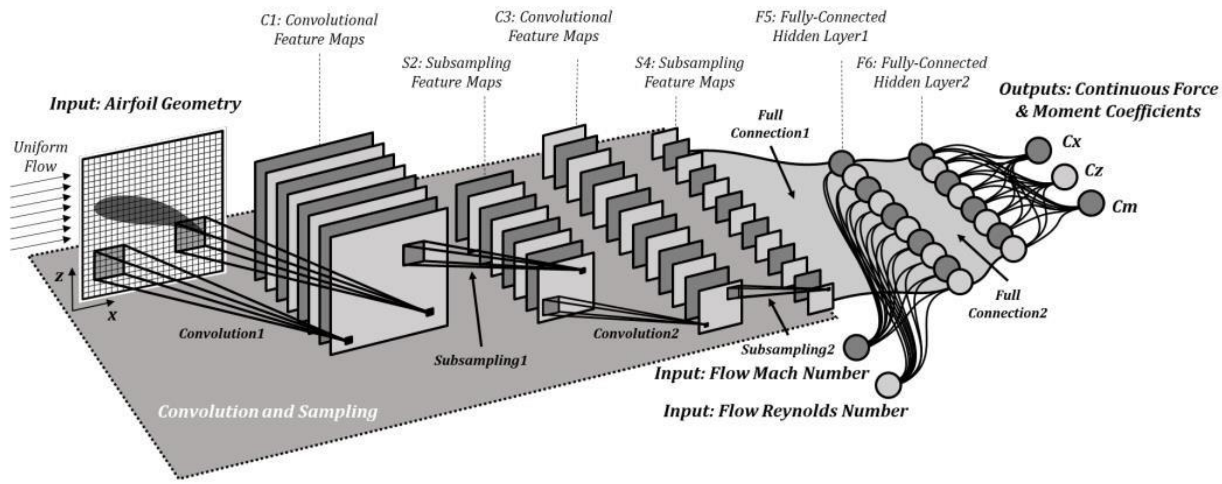

Figure 4: Digitized LetNet-5 for aerodynamic modelling tasks

LetNet-5 is a convolutional neural network architecture that was designed for image classification tasks. The LetNet-5 architecture consists of seven layers which are two convolutional layers, two subsampling layers, and three fully connected layers. LetNet-5 was groundbreaking at the time of its introduction because it demonstrated the potential of deep learning for image classification tasks. As a conclusion of this paper, the study demonstrates the potential of CNN for predicting airfoil lift coefficient which could have important aspects for aircraft design and optimization. By using the power of deep learning this approach could lead to a more accurate and efficient method of drag and lift coefficient prediction by improving the performance and safety of the aircraft. Here they used Adadleta algorithm to train their neural network. This Adadelta algorithm gives more error reduction with notable insensitivity to the choices in hyperparameters for weight optimization.

[8] In this paper they use the convolutional neural network for approximating steady flow fields. Here in this paper, it is mentioned that accurately predicting flow fields is essential for many engineering applications such as aerodynamics and fluid dynamics. Here is the paper proof that using CNN model to predict the steady flow field based on the boundary conditions of the problem. They show their performance of their CNN model on several important problems in fluid dynamics comparing it to traditional numerical methods such as finite element analysis and finite volume analysis.

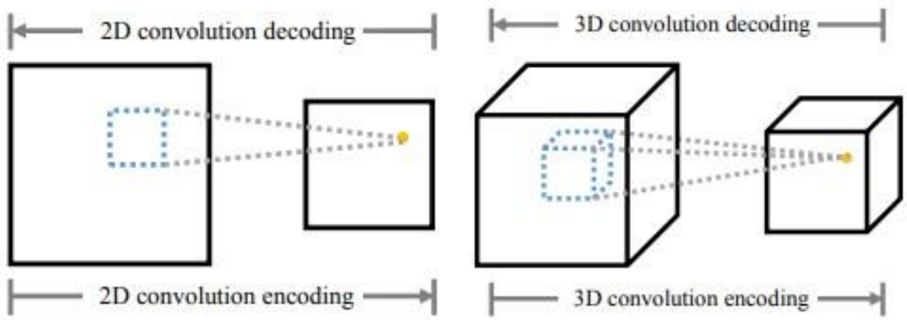

Figure 5 :2D and 3D convolutional encoding & decoding

Convolutional encoding is a process that takes a sequence of input bits and produces a longer sequence of output bits by applying a convolutional code. (A convolutional code is a type of error-correcting code used in digital communication systems to add redundancy to a data stream to detect and correct errors that may occur during transmission).

The major impact of this type of research is to make visual predictions of the steady flow around shapes and real-life shapes (ex: - cars). The conclusion of this paper is that it says study demonstration of the potential of CNN for the approximating steady flow fields which could have important aspects for engineering applications.

[8] This paper is based on the topic that ‘’An efficient deep learning technique for the Navier stokes equations: Application to unsteady wake flow dynamics’’. The Navier stokes equations describes the motion of the fluid flow. Authors are proposed here modern technology that combines deep learning with traditional numerical methods to efficiently solve the Navier stocks equations. Majorly this paper uses convolutional neural networks to learn the complex relationships between the inputs and outputs of the numerical simulations for training, validation, and testing data. Here the used CNN was trained using huge amount of dataset of numerical simulations and once trained it accurately predicted the solutions to be the Navier stocks equations for new unseen inputs.

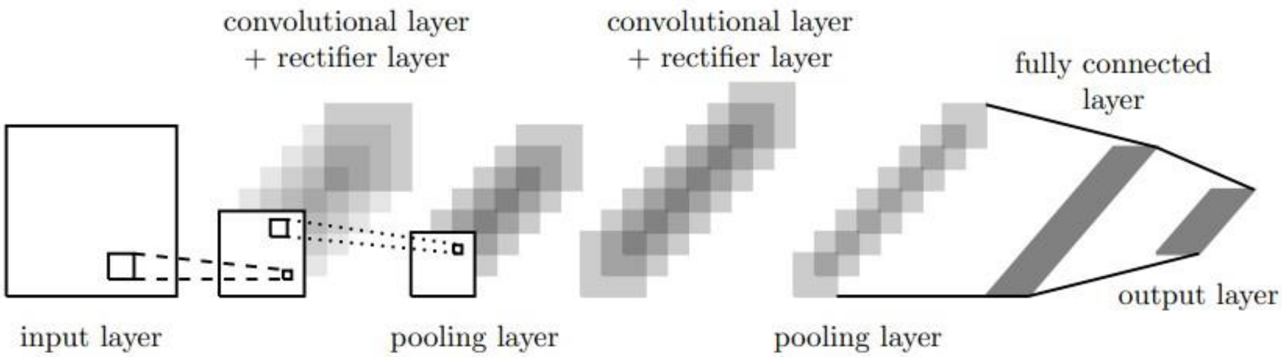

Figure 6: Representation of convolutional neural network architecture

This paper demonstrates the effectiveness of their work by applying it to the problem of unsteady wake flow dynamics. Not only that, compare their approach to traditional numerical methods and how their technique is faster and can make accurate predictions. Their technique can also be used for real time control of fluid flow which has important applications in range of fields including aerospace, automobile engineering and engineering productions etc.

Deep convolutional networks for large scale image recognition were another paper that has been given important key factors of neural network image recognition. This presents a deep learning architecture called visual geometry group (VGG) network. The VGG network is a one of deep convolutional neural network that achieves state of the art performance on the image data set feature and when the data set is large and large image scale image recognition dataset that contains over a million labelled images. The VGG technical architecture consists of series of convolutional layers which are followed by max pooling layers and then several fully connected layers. Here the convolutional layers are used for extracting the features form the large, scaled image input and the max pooling layers are used to down sample the feature maps to reduce the partial dimensionality that large, scaled input image. The fully connected layers are used for classification in this paper. The one key role of this VGG network is the use of exceedingly small 3*3 convolutional filters throughout the VGG network. By using small filters, they developed some advantages. They are to capture more fine-grained details in the input mage which can lead to improved classification performance. The authors also here with different networks depths and find that increasing the depth of the network leads to better performance in the ImageNet dataset. Not only that the VGG network gained state of the art performance on the Image net dataset at the time its publication and its architecture has since become a widely used mark for deep learning models in computer vision and the computer neural network.

Another paper views which were done based on the deep residual learning for image recognition. This paper proposes a deep neural network architecture which is called Resnet (Residual network). The ResNet which was designed to overcome some major problem. This problem is vanishing gradients in very deep neural networks, which can make it difficult to train such networks effectively and efficiently. When using ResNets it gives some advantages such as ResNets can be deeper than neural network architecture as they allows information to bypass certain layers of thee network through shortcut connections , It has been shown to state art performance on variety of computer vision tasks such as image classification and object detection, ResNets also to converge faster during training than traditional neural networks which means it has good performance with less time and efficient, It has also faster learning rate than traditional neural networks. In ResNets which allow the network to learn residual function with respect to identity mapping. In can be explored as that instead of trying to directly learn the desired mapping between input and output network learns the difference between Input and the output after that adds this difference back to the input to produce output. This approach makes it easier for the networks to learn the desired mapping. The authors experiment with different depths of the ResNet architecture and find that increasing the depth of the network leads to improved performance on the ImageNet dataset. They also compare the performance of the ResNet to other state-of-the-art deep learning models and show that the ResNet achieves better accuracy with fewer parameters.

METHODOLOGY

Predicting the drag coefficient of two-dimensional shapes is an important problem in fluid mechanics, as it can help optimize the design of various objects, such as aircraft, automobiles, and ships. Machine learning can be used to develop predictive models that can accurately estimate the drag coefficient of arbitrary two-dimensional shapes. Here the random shapes were generated from using softr.oi online shape generator.

Figure 7: Randomly generated two dimensional shapes.

One approach to developing a machine learning-based drag coefficient prediction model is to use a dataset of shapes with known drag coefficients to train a neural network. The dataset can be generated using CFD simulations, where the drag coefficient is calculated for each shape. The neural network can then be trained to learn the relationship between the shape geometry and the drag coefficient.

In this research the main objective is to make a model from using machine learning algorithm to predict the drag coefficient of random two-dimensional geometry. This is the main objective of this research.

1. Develop a convolutional neural network (CNN) for drag prediction.
2. Train the model with high accuracy.
3. Compare tabulated drag coefficients with predicted drag coefficients. 4. Predict drag coefficient for a given custom image.

The major purpose of this research is,

1. Designers can use this machine learning model to predict drag coefficients of newly designed shapes.

   Ex : newly designed helmet shapes

   automobile newly added shape drag coefficients.

2. Designing creative shapes.
3. Machine learning models can be used for AI based product designs.

When a two-dimensional object is placed inside a free stream flow, two forces act on that object. Here we only consider the drag force and its coefficient only. The magnitude of the force varies with the geometry as well as the flow conditions such as velocity (v), Reynold's number (Re). To represent those two forces dimensionless parameters which are known as drag coefficient (C_D) are used.

$$C_D = \frac{2F}{\rho U^2 A}$$

Were,

C_L = Coefficient of lift

C_D = Coefficient of drag

A = Reference area / Projected area

ρ = Density of fluid v = Fluid

speed F_D = Drag force

Reynold's number is defined as follows,

$$R_e = \frac{\rho UL}{\mu}$$

Where, ρ = Density

of fluid.

U = Freestream velocity.

L = Characteristic length.

μ = Dynamic viscosity of fluid.

In this research, the first goal is to create the data set before the training of the machine learning model. To get the drag coefficient values I followed solid workflow simulations for a two-dimensional case. First, I created the image into a 3D object. After that for the computational domain I selected domain length which is equal to the extruded length of the object.

In the next step the parameters were set up of the flow. This is an external flow and air was selected as the flow medium. For the flow condition was selected as laminar flow region because here we maintain the Reynolds number at 10. Here the selected random shapes were not in the same horizontal length because of that the velocity value will vary from shape to shape to maintain a constant Reynolds number value as 10.

CONVOLUTIONAL NEURAL NETWORK

Convolutional neural networks (CNNs) are a type of neural network commonly used in computer vision tasks such as image recognition. They are particularly well-suited for these tasks because they can learn hierarchical representations of images. In a CNN, the convolutional layers use local receptive fields to learn features from small patches of an image, and the weights for these features are shared across the entire image. This allows the network to learn spatially invariant features that can be used to recognize objects regardless of their position in the image. By stacking multiple convolutional layers, the network can learn more abstract and complex features that are able to distinguish between different classes of objects.

When working with random images as our input we must use convolutional layers instead of fully connected layers. Because the fully connected layers most of the times use with less common image recognition tasks. Here each convolutional kernel which is inside the convolutional neural network will be used for the purpose of extracting specific features from the input random shape image. During the forward processing, each kernel involves with successfully match with input image to generate an activation map which shows the response of the kernel at every spatial position in the given input. This flow of the process will allow the network to learn spatially invariant features that can be used to identify the suitable and correct object and patterns that produce the input image.

Most of the times convolutional layers are followed by pooling layers. Pooling layer major advantage is to reduce the spatial size of the feature maps generated by the convolutional layers while retaining the important spatial information. And it can decrease the number of degrees of freedom in the neural network and make efficient procedures. Not only that pooling layers can spread the initial date throughout the successive convolutional layers. When it comes to CNN it gives some of the major advantages such as shared weights, efficient detection of spatial features of the given image. For the selected data they were split into three categories. Those are training set, validation set, test set. In training data set is used to train the model and update its parameters. In validation set is used to monitor the accuracy of the model during the training and make some decisions about the hyperparameters and the network architecture. Finally, the test set is used to evaluate the final performance.

of the model that on did not train data after the training is complete. Here we need to remove the dataset from overfitting problem. If overfitting happens, the model performs well on the training data and poorly on unseen data. Another factor is the validation and test sets should not overlap with each other. If it happens this could bias the evaluation of the machine learning model performance. For the first 100 shapes were put into following neural network and those are the basic layers of CNN.

ACTIVATION FUNCTION

In this convolutional neural network, it is an essential component, and it gives nonlinear into the model and allowing the model to learn complex patterns and relationships that already in the dataset. Each neuron in a neural network applies an activation function to its input image and output is then passed to the next layer of neurons.

In recent machine learning techniques use Rectified linear unit (ReLU) use for hidden layers in neural networks for classification tasks due to its complexity and computational efficiency. In this training process we must reduce the loss function also. The loss function measures the difference between the predicted output of the neural network and the true output. The learning process objective is to minimize the difference between which is achieved by updating the networks parameters. Finally, the process of adjusting the parameters is repeated over many training inputs or they are also called epochs. Here we do this process until the loss function is minimized to an acceptable level. At this point we can decide to the neural networks can be used to make predictions on new unseen data as well.

As the first stage of training section, we used 100 input data. The low amount of data in our training section can lead to overfitting the neural network model. A small network with few parameters may not have enough strength to learn the underlying patterns in the data and its features accurately and may perform poorly on both training and validation set. As an implementation of neural network, we use this amount data and after the neural network making the input data will increase up to more than 1000 inputs. In further model training stages, we should limit the overfitting. From gathering more data, reducing the size of the network can lead networks to lower overfitting.

For the neural network implementation stage, we use inbuild libraries which is already with TensorFlow. Here used architecture consist of a pattern of repeated convolutional and pooling layers with varying number of filters in each layer. (These filters or kernels are used to extract features from the input image. They are small matrices that slightly vary over the input image and perform element wise multiplication and summation to produce feature map). Here includes two convolutional layers with initial layer including 8 filters. Ans the network is terminated with two dense layer of size 16 and last layer used for outputting predicted drag coefficient using linear activation function.

DATA PREPROCESSING

For the machine learning image processing task there are several important libraries that were used here. For the data manipulation panda's library was used. TensorFlow was used which is a popular open-source library for machine learning and NumPy helps to large multidimensional arrays and mathematical functions. In the data processing section 12000 images and drag coefficient values were obtained in the previous research paper. [9]

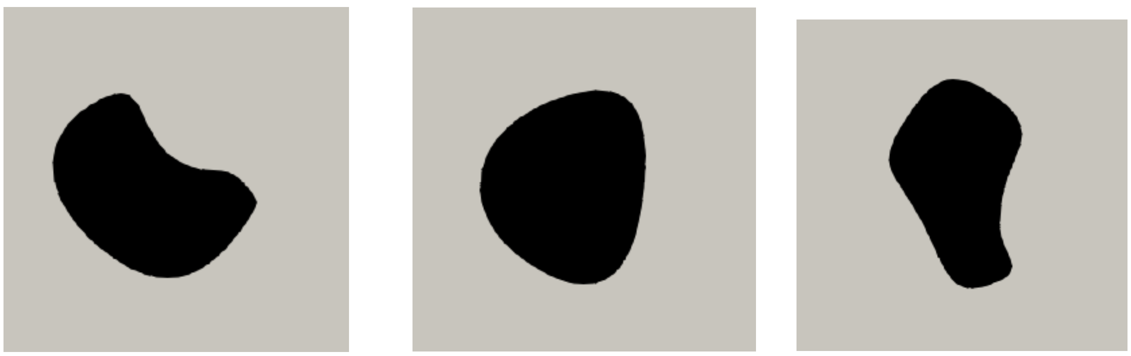

Figure 8: Generated two dimensional shapes.

.

Tha data set was split into two categories which are trained set and the test set. For the trained set 10200 images were trained using convolution neural networks and 1800 data were used as the test set. Then the 20% of the trained data set was split to create the validation dataset. The purpose of this validation set is to evaluate whether the convolutional architecture which was used in training is correct or wrong. After that, every image was rescaled 0 – 1 range to increase the processing speed of the data set. Then the images were converted into greyscale to reduce the complexity of the images Because greyscale images have a single set channel which they have low memory requirements and computational power compared to the RGB images that consist with three channels. From converting RGB images into greyscale ones the task can focus on the overall brightness of the image which easily extracts the features of a particular image.

For the images loading and training, validation and testing were done by using special functions. For efficient data loading data generators were used and those load image data from specified data frames. These data generators fetch the images in batch wise and allow the model to process and reduce memory requirements and allow for efficient training. When processing the images as batches model can vary parallel processing and GPU acceleration causes to more faster training and predicting times.

For the architecture for this 2D image drag prediction task plays a key role here. For the regression task convolutional neural network architecture was on grayscale images. For the input layer model performs grayscale images with dimensions of 256*256 pixels. After the feature extraction multiple convolutional layers were applied from the input images. Each layer of these layers applies a set of filters (16,32,64) to the input by providing ReLU activation function. To reduce the spatial dimensions of the output Max pooling layers were used in this architecture. For the output layer was flattened into a 1D vector to prepare the fully connected layers. To learn the complex relationships between extracted features dense layers were added to the architecture. Finally, the fully connected layer only has a single unit and here uses linear activation function for predicting the regression target value.

For the future works of this research can be conducted as follows.

1. Generate more random shapes. (More than 12000)
2. Training the neural network. (With more than 12000 data)
3. Developed to 3D shapes drag prediction.
4. Finalized the network with an application.
5. Experimentally testing of real-life shapes

RESULTS AND DISCUSSION

Tested architecture 01

After generating random shapes all the drag coefficient values were found using simulations. After that we CNN model training was done using TensorFlow libraries. The following shows the tested convolutional neural network architecture. Here we used 3×3 conv,16 layer, it means that convolutional layer with 16 filters and each filter is a small matrix of size 3×3 . When we use multiple convolutional layers, it helps to enable the network to learn hierarchical representation of input image. Here each filter will make a separate feature map. Here we use a 2×2 pooling layer and it refers to layer that reduces spatial dimensions of the feature map.

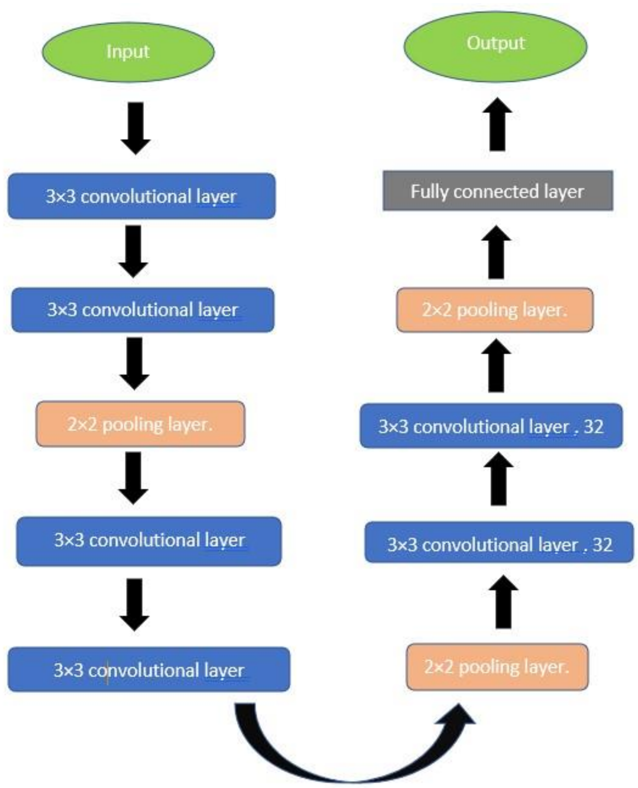

Figure 9: Architecture used CNN architecture for drag coefficient prediction.

For the data preprocessing part, the data frame was read (random shape and its drag coefficients form excel sheet) and tabulates them as shape id value. After that, the image path was assigned to an address. The whole data frame was split into 70 for the training and 30 for the testing. For validation 20% was selected from the training dataset. From validation the main purpose is to tune the accuracy of the network. Then rescale the image pixel values between 0-1 to improve the processing speed of the network and reduce the RGB color image to a grayscale image also to improve the processing speed of the neural network.

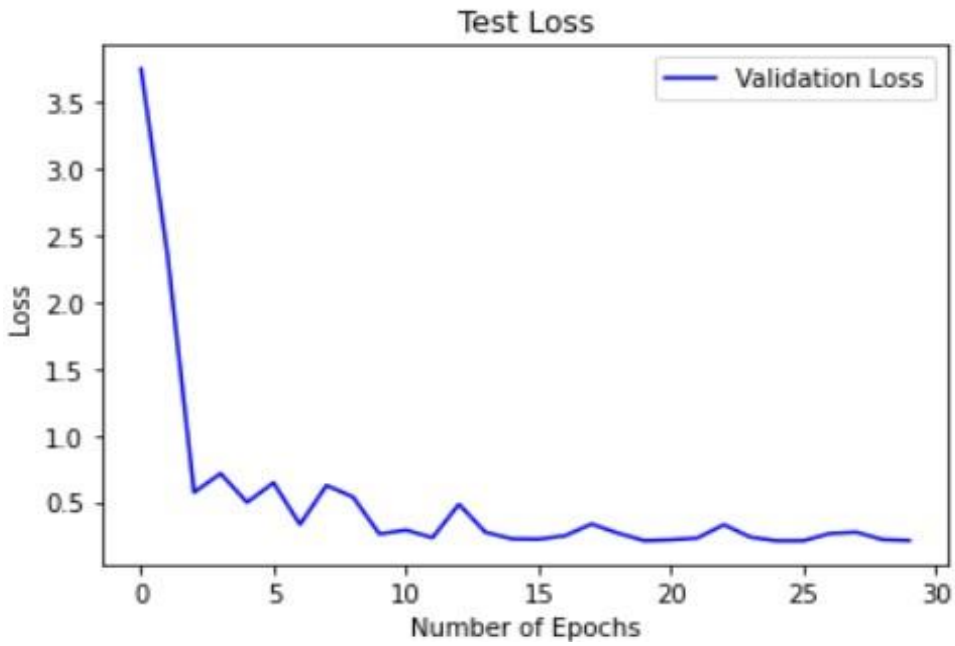

Figure 10: Training loss with the initial architecture

For this tested CNN architecture was given a better accuracy and lower loss of the generated 100 data. For these data amount the lower loss given at the 28 epochs.

Table 1: Relative error table for some shapes

Shape	Exact drag coefficient	Predicted Drag coefficient	Relative error
Shape 66	2.54185	2.27895	10.34 %
Shape 89	2.18827	2.30818	5.4%
Shape 62	2.1859	2.06021	5.7%
Shape 55	2.11403	2.0505	3%

The main drawback of this network was, we only used only 100 data and those are not enough to train the CNN network in better accuracy. Here our relative error results are between 3% - 10% of the total. These values are higher and in the remaining works we need to increase data amount to get better accuracy of the predicted drag coefficients.

After the first training process the CNN training process was done with 12000 images and its drag coefficient values with the same architecture. After training all the images consider shape_11825 for evaluating the architecture. The actual drag coefficient value for the shape is 1.8433. Here considers the predicted value with actual value of the shape and calculated its relative error. The goal of this comparison is to find out the optimum number epochs with lowest validation loss.

Table 2: Number of epochs comparison with shape_11825

Number of epochs	Predicted Drag coefficient	Validation loss	Relative error (%)
6	2.0350	0.3117	10.39
7	2.12320	0.3090	15.18
8	2.071	0.3080	12.35
9	2.0258	0.3129	9.9
10	2.0496	0.3101	11.19
11	2.1088	0.3084	14.4
15	2.0731	0.3085	12.46
20	2.0964	0.3081	13.73
30	2.1400	0.3100	16.09

This shape was included in the test set of the data frame. According to the table when number epochs is 9 the predicted drag coefficient value was given lowest relative error. But this value was high. As the result of this comparison got to know that the CNN architecture needs to develop to get more accurate results.

Tested Architecture 02

The prediction values of architecture 01 did not reach the desired output values. Because of that architecture was changed and checked the beast fit architecture with comparing the validation loss of the tested architecture.

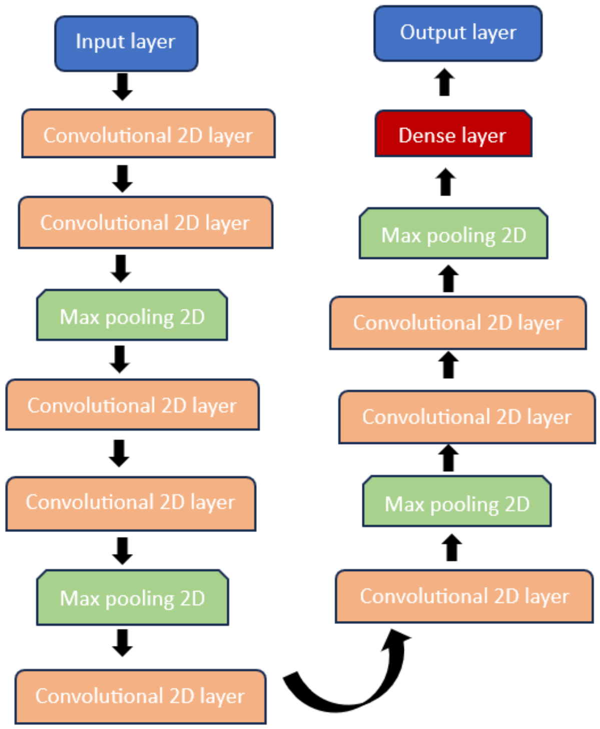

Figure 11: Tested neural network architecture 02.

In this architecture was given validation loss as 0.3142. With this value the prediction was not perform the accurate results.

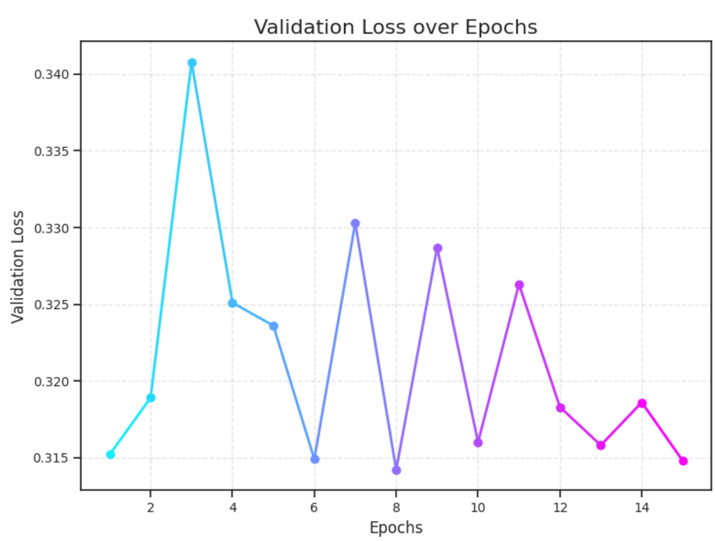

Figure 12: Validation loss graph for architecture 02

Tested architecture 03

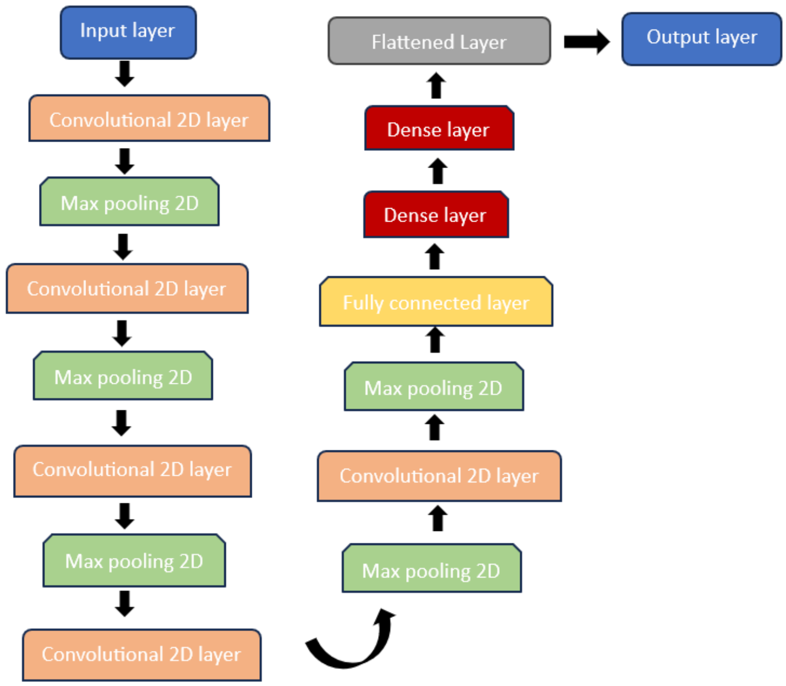

Figure 13: Tested neural network architecture 03.

In this architecture was also given validation loss as 0.021. According to this value the model was not predicted values but not accurate.

Figure 14: Validation loss graph for architecture 03

Tested architecture 04

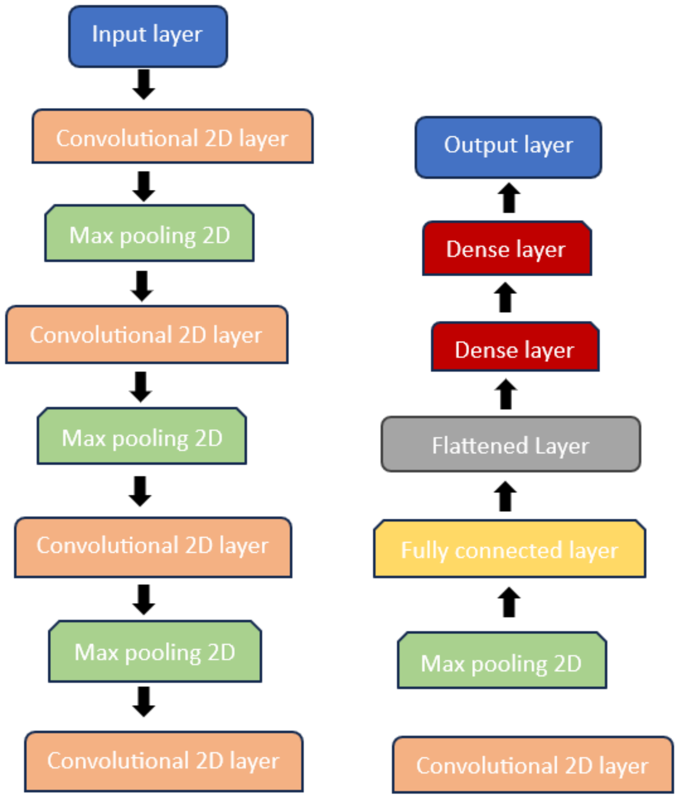

Figure 15: Tested neural network architecture 04.

According to the architecture 04 the minimum validation loss was performed as 0.024. With this architecture the drag coefficient prediction model was given better results with higher accuracy than the above tested architectures.

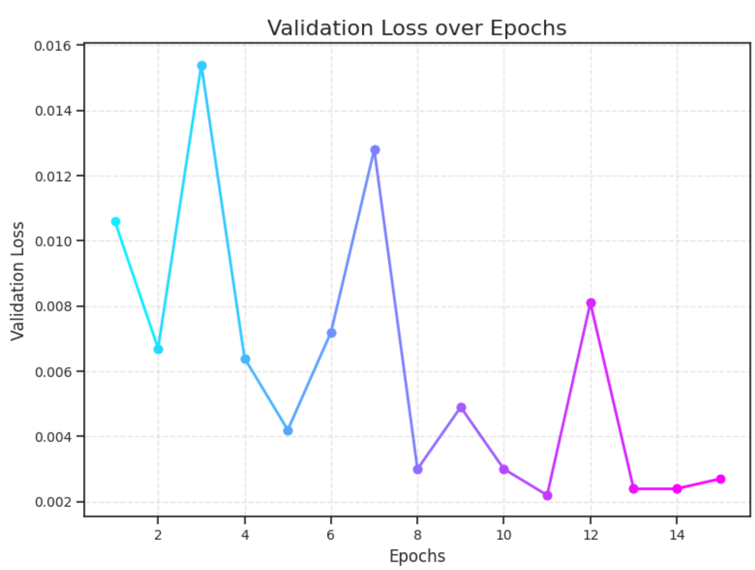

Figure 16: Validation loss graph for architecture 04

Tested architecture 05

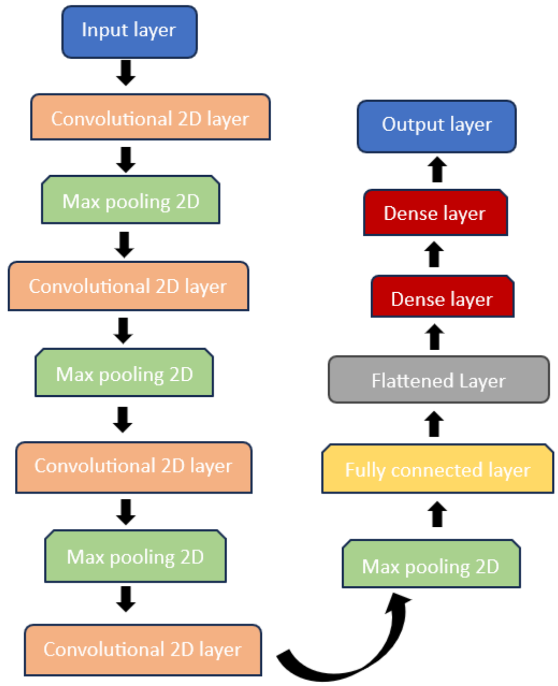

Figure 17: Tested neural network architecture 05.

For greater accuracy and the best results, the architecture was little bit changes with the architecture 04. But here this newly developed architecture did not perform well with the shapes and predictions were not accurate. Validation loss was in this architecture 0.2381.

Figure 18: Validation loss graph for architecture 05

Tested architecture 06

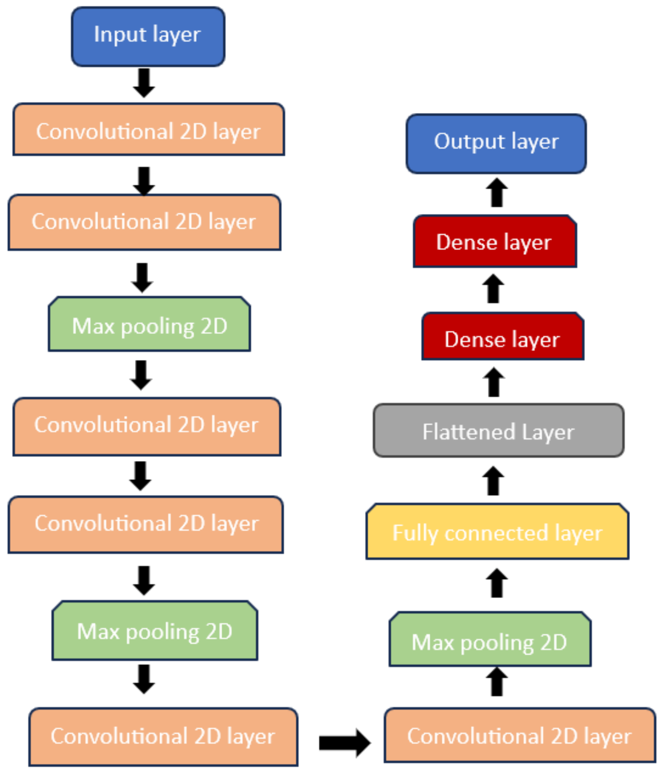

Figure 19: Tested neural network architecture 06.

In architecture 6 also a change stage of the architecture 04. But here also the validation loss was 0.2991. With this value the prediction results were not accurate.

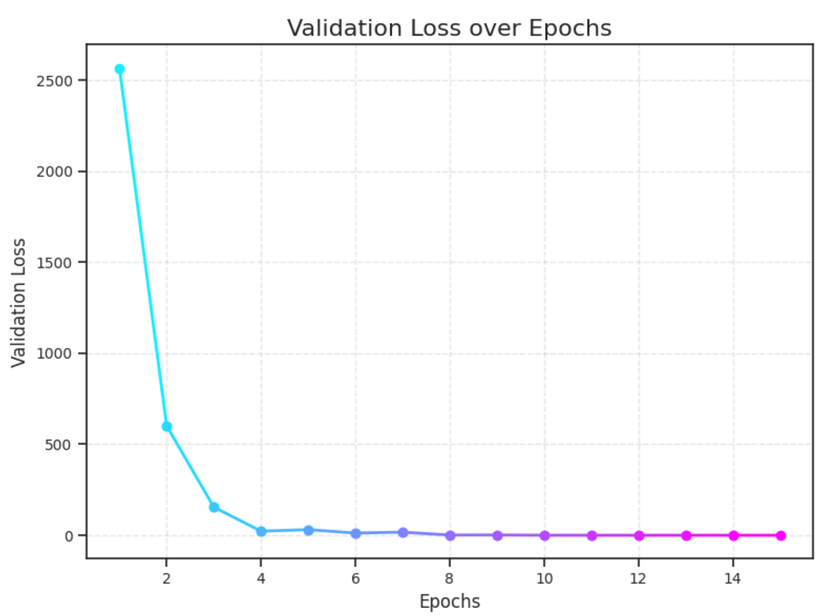

Figure 20: Validation loss graph for architecture 06

Tested architecture 07

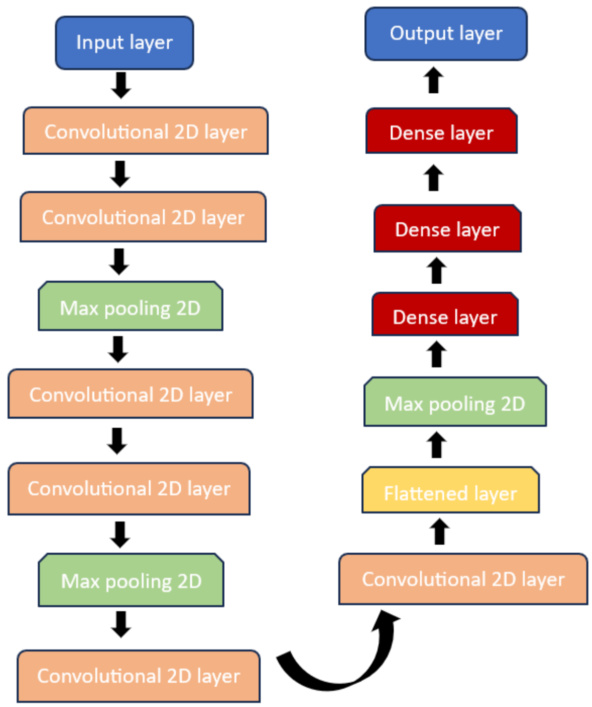

Figure 21: Tested neural network architecture 07.

In this architecture the validation loss was 0.0017 Which was the minimum validation loss compared to the other above 6 architectures. Finally, this architecture was selected as the higher accuracy architecture and with this the prediction of drag coefficients of two-dimensional images were given higher accurate results.

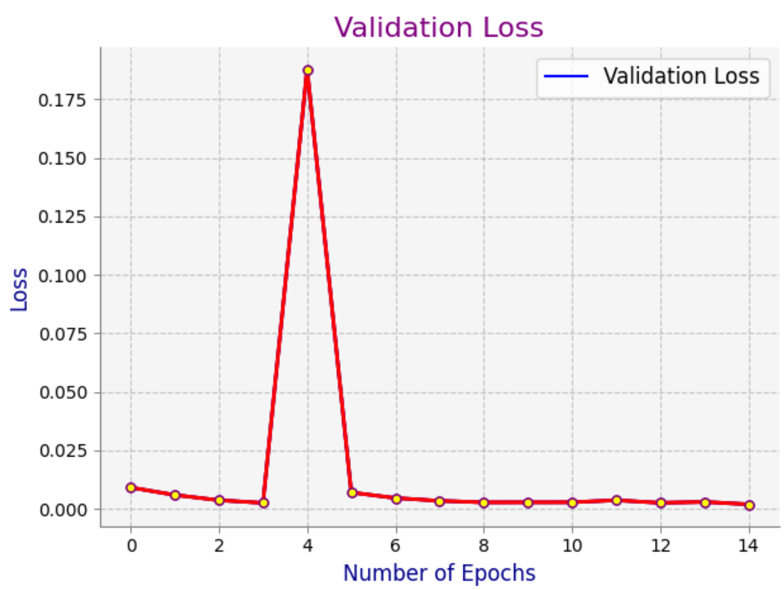

Figure 22: Validation loss graph for architecture 07

Table 3: Validation loss comparison with tested architectures

Architecture No	Minimum Validation loss
1	0.3012
2	0.3142
3	0.021
4	0.0024
5	0.2381
6	0.2991
7	0.0017

This table shows the comparison of 7 architectures which were done in this research. From the comparison the 7th architecture was selected as convolutional neural architecture for this machine learning model.

Optimum number of epochs

For the model performance determining number of epochs helps to the model for without underfitting and overfitting. Not only that optimum number of epochs also help to reduce the time and resource efficiency, early stopping, and give the model to robustness. Finding the number of epochs is essential for achieving the best possible model performance for the drag prediction of given image. For above architecture 01, when increasing number of epochs validation loss was remains in a large value and relative error was also high.

Table 4: Number of epochs, Predicted value , Validation loss , Relative error graph for architecture 01.

Number of epochs	Predicted Drag coefficient	Validation loss	Relative error (%)
6	2.0350	0.3117	10.39
7	2.12320	0.3090	15.18
8	2.071	0.3080	12.35
9	2.0258	0.3129	9.9
10	2.0496	0.3101	11.19
11	2.1088	0.3084	14.4
15	2.0731	0.3085	12.46
20	2.0964	0.3081	13.73
30	2.1400	0.3100	16.09

- Finally, the 7 th architecture was given more accurate predicted values with lowest relative error. According to the below table when the model reaches the 15 number of epochs the model performs lower validation loss and lower relative error for this model.

Table 5: Number of epochs, Predicted value, Validation loss, Relative error graph for architecture 07.

Number of epochs	Predicted Drag coefficient	Validation loss	Relative error (%)
6	2.65	0.0025	1.5
7	2.7763	0.0032	2.32
8	2.7416	0.0022	1.04
9	2.6716	0.0026	1.5
10	2.2.65	0.0038	2.33
15	2.7062	0.0017	0.26

Relative error vs batch size

Comparing relative errors with the different batch sizes can help to increase the model performance of the model. It shows how well models are learning with different batches. Not only that, but the batch size is also a hyperparameter that influences the training process. Finding the relative error for different batch sizes helps to find optimal batch size of this CNN model. For memory and computation efficiency also increase with the optimal number of batches. overfitting happens because the model performs well on the training data but fails in the unseen custom data. From analyzing the relative error and its lower range across different batch sizes we can observe the model's behavior.

- Optimal batch size can be found out from this graph.
- The hyperparameter tuning and understanding of the model behavior is another advantage of relative error and batch size graph.

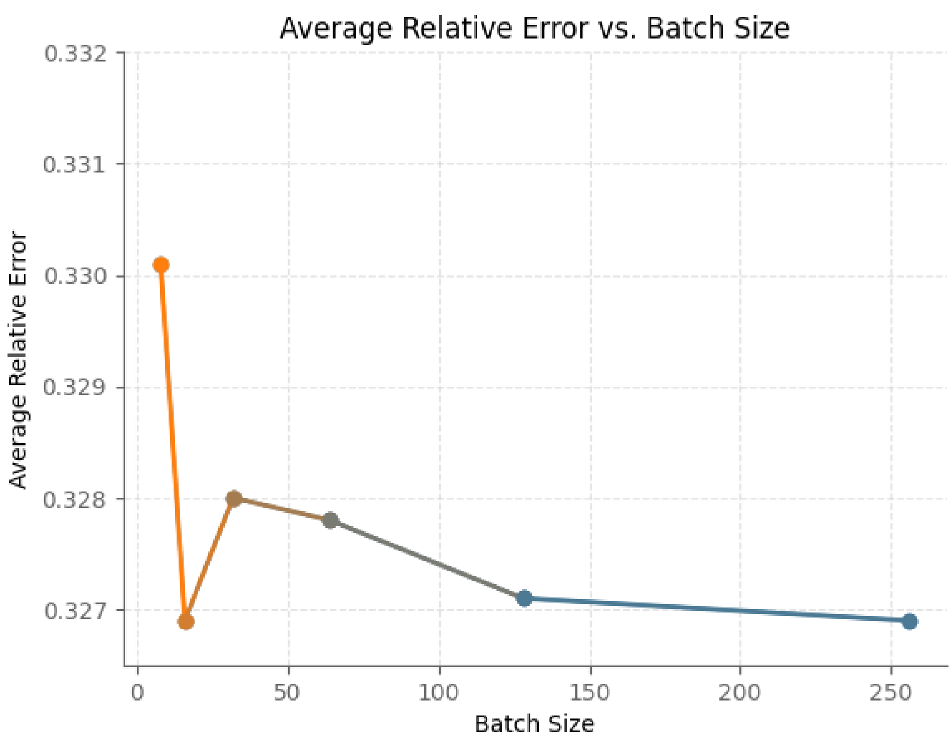

Figure 23 : Average relative error vs batch sized graph

Real life examples and drag coefficient predictions.

Applications: -

1. Automobile industry, drag prediction is valuable in designing shapes of the cars to reduce the aerodynamic drag.
2. Swimming and cycling reducing drag will affect to increase the performance.
3. Architects and civil engineers can utilize 2D drag predictions to give better performance of design buildings and structures to decrease the wind resistance.

Table 6: Drag coefficient prediction for custom images.

Shape	Actual drag coefficient	Predicted drag coefficient	Relative error (%)
	1.589	1.61	1.32
	1.764	1.765	0.056
 NACA 4424	1.263	1.298	2.77
 NACA 4412	1.121	1.125	0.35
 NACA 0018	1.186	1.190	0.33

CONCLUSION

Machine learning techniques are widely used present in computational fluid dynamics to improve accuracy and performances better than the traditional computations. Here this research was done to predict the drag coefficients for two dimensional geometries using convolutional neural networks. The developed CNN architecture was used to extract the features of the images. Finally, the predicted values form the neural network and simulation values were compared and relative error also compared. To develop the accuracy of the neural network it is necessary to increase the number of data points and tune the hyperparameters of the neural network. There are some major applications based on two-dimensional drag prediction in engineering and physics. Predicting generated drag forces on airfoils and wings is important in aircraft design. From considering the flow characteristics and properties around two dimensional airfoils experts can understand the drag coefficient and optimize the shape for improved aerodynamic performance. When considering the automobile industry, drag prediction is valuable in designing shapes of the cars to reduce the aerodynamic drag.

To improve fuel efficiency engineers can optimize the shape of automobiles by using this kind of neural network drag prediction method. There are many fields that the drag will crucially affect. In sprots such as swimming and cycling reducing drag will affect to increase the performance. From considering flow around 2D shapes and models of sports equipment like swim kits, sky suits can help designers optimize their shape with minimizing the drag. Not only architects and civil engineers can utilize 2D drag predictions to give better performance of design buildings and structures to decrease the wind resistance. This will majorly affect the tall buildings and bridges where high drag forces can lead to structural failures of the buildings and bridges. Designing creative shapes and AI based design will be improved from using this kind of drag prediction method.

The neural networks hyperparameter tuning helps to optimize the settings that control its CNN architecture and the model training process. Learning rate, batch size, networks depth , and the kernel size can be identified as the hyperparameters of the model .The experiment can be done with different values of these hyperparameters to find out the efficient combination that helps to improve the model with best results in terms of prediction accuracy of the CNN model. From iteratively tuning the CNN model and increasing the dataset we can gradually improve the accuracy of the neural networks predictions for drag coefficients in CFD simulations. The continuous iterative process for continuous learning and refinement of the model will allow the model to make reliable predictions.

The base code for this Neural network for the two dimensional drag prediction :- **https://colab.research.google.com/drive/1BgbVxsDx0GOJPnGtFcCV8AVfNbrnkMSU?usp=drive_link**

REFERENCES

1. “Damassets.autodesk.net.” [Online]. Available: https://damassets.autodesk.net/content/dam/autodesk/research/publicationsassets/pdf/convolutional-neural-networks-for.pdf. [Accessed: 28-Feb-2023].

2. T. P. Miyanawala and R. K. Jaiman, “An efficient deep learning technique for the navier-stokes equations: Application to unsteady wake flow dynamics,” *arXiv.org*, 15-Aug-2018. [Online]. Available: https://arxiv.org/abs/1710.09099. [Accessed: 28-Feb-2023].

3. “Application of convolutional neural network to predict airfoil lift ...” [Online]. Available: https://www.researchgate.net/publication/322313274_Application_of_Convolutional_Neural_Network_to_Predict_Airfoil_Lift_Coefficient. [Accessed: 28-Feb-2023].

4. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv.org*, 10-Apr-2015. [Online]. Available: https://arxiv.org/abs/1409.1556. [Accessed: 28-Feb-2023].

5. “Deep residual learning for image recognition - cv-foundation.org.” [Online]. Available: https://www.cvfoundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_20 16_paper.pdf. [Accessed: 28-Feb-2023].

6. “Machine learning methods for turbulence modeling in subsonic flows over ...” [Online]. Available: https://www.researchgate.net/publication/325817088_Machine_learning_methods_for_turbulence_modeling_in_subsonic_flows_over_airfoils. [Accessed: 28-Feb-2023].

7. “Application of artificial neural networks to the design of ...” [Online]. Available: https://arc.aiaa.org/doi/10.2514/2.5725. [Accessed: 28-Feb-2023].

8. W. Peng, Y. Zhang, E. Laurendeau, and M. C. Desmarais, “Learning aerodynamics with neural network,” *Nature News*, 26-Apr-2022. [Online]. Available: https://www.nature.com/articles/s41598-022-10737-4. [Accessed: 28-Feb-2023].

9. “A convolutional neural network approach to training ... - researchgate.” [Online]. Available: https://www.researchgate.net/publication/318142951_A_Convolutional_Neural_Network_Approach_to_Training_Predictors_for_Airfoil_Performance. [Accessed: 28-Feb-2023].

10.j. k. ,. a. ,. v. ,. i. g. josef musil, "Towards sustainable architecture 3D convolutional neural networks for compuatational fluid dynamics simulation and reverse design workflow," 2016.

11.s. j. j. ,. v. o. s. ,. a. A. .. a. ,. j. v. g. t. ,. b. g. Paulo alexandre costa rocha, "Deep neural network modeling for CFD simulations banchmarking the fourier neural operator on the lid driven cavity case," 2022.

12.Y. w. ,. x. y. y. ,. z. h. c. ,. w. t. w. ,. n. a. Ming yu wu, "fast prediction of flow field around airfoils based on deep convolutional neural network," 2015.

13.a. h. ,. a. k. Matthias eichinger, "surrogate convolutional neural network models for steady compuatational fluid dynamics simulations," 2020.

14. y. w. ,. a. m. ,. j. l. ,. x. c. xiaohui yan, "CFD-CNN modeling of the concentration field of multiport buoyant jets," 2022.

15.k. f. ,. k. z. ,. a. G. .. N. ,. k. f. masaki morimoto, "Convolutional neural networks for fluid flow anlysis toward effective metamodeling and low dimensionalization," 2021.

16.a. h. ,. a. k. Matthias eichinger, "surrogate convolutional neural network models for steady computationsl fluid dynamics simulations," 2022.