Scientific Research Publishing

World Journal of Mechanics, 2024, 14, 169-184
https://www.scirp.org/journal/wjm
ISSN Online: 2160-0503
ISSN Print: 2160-049X

Study on Predicting the Mode Development Process of Taylor Vortices Using Convolutional Neural Networks

Hiroyuki Furukawa, Takeomi Yamazaki

Department of Mechanical Engineering, Meijo University, Nagoya, Japan
Email: furukawa@meijo-u.ac.jp

How to cite this paper: Furukawa, H. and Yamazaki, T. (2024) Study on Predicting the Mode Development Process of Taylor Vortices Using Convolutional Neural Networks. *World Journal of Mechanics*, **14**, 169-184.
https://doi.org/10.4236/wjm.2024.148008

Received: August 1, 2024
Accepted: August 27, 2024
Published: August 30, 2024

Abstract

The study investigated Taylor vortex flow between rotating double cylinders using a convolutional neural network (CNN). By combining numerical results of vortex flow for specific periods after vortex onset, the researchers aimed to determine if mode discrimination was possible in the combined images. They used images taken at various intervals: 20 images at 1 second, 30 images at 1.5 seconds, 40 images at 2 seconds, 50 images at 2.5 seconds, 60 images at 3 seconds, and 67 images at 3.35 seconds after vortex onset. The goal was to compare the accuracy rates in predicting the mode development process of the vortex. The study concluded that the mode development process of the Taylor vortex can be discriminated by combining images taken at specific time intervals after the vortex occurs and training the CNN with these images as teacher data. The results showed that the most efficient prediction of the mode development process was achieved when 50 images taken at 2.5 seconds were used for learning. This highlights the potential of using CNNs in fluid dynamics research, specifically in analyzing and predicting the behavior of vortex flows.

Keywords

Taylor Vortex Flow, CNN, CFD

1. Introduction

Image recognition technology based on convolutional neural networks has been used to improve quality and detect abnormalities in many industrial processes, such as manufacturing, quality control, and maintenance of industrial products, and its applications are expanding. In this context, the prediction of the mode development process of Taylor vortices generated in journal bearings is

a particularly noteworthy issue. Journal bearings are part of the machine elements and play an important role in mechanical devices such as engines and pumps, and their accurate operation has a direct impact on the life and performance of the machine.

Taylor vortex flow is a fundamental phenomenon in fluid dynamics that is crucial for understanding fluid behavior, rotational motion, and flow stability [1]. It helps in understanding nonlinear flow patterns, aiding in the study of nonlinear dynamics [2]. As applications, Taylor vortices are used in rotating reactors for efficient chemical reactions [3], improving fuel cell efficiency, and in research on blood circulation and drug distribution. The study of Taylor vortices has potential applications in various fields and is expected to continue developing.

Recently, the following research on Taylor vortex flow has been carried out. Zhijie Wang, *et al.* [4] present a two-stenosis aorta model to mimic vortical flow in vascular aneurysms. Using computational fluid dynamics (CFD) simulations based on physiological and anatomical data from adult rabbits, the research aims to replicate disordered eddy flows in aneurysms. Dong Liu *et al.* [5] explore the effects of slit walls and heat transfer on Taylor vortex flow. Researchers from Jiangsu University and Gyeongsang National University conducted the study, focusing on how slit walls and heat transfer influence the flow dynamics in an annular gap between concentric cylinders. Zixuan Yang *et al.* [6] analyze the sustaining mechanisms of Taylor-Görtler-like vortices through direct numerical simulation. The study focuses on understanding how these vortices are maintained in turbulent channel flows subjected to fast rotation. Prashanth Ramesh and Meheboob Alam [7] investigate the coexistence of spiral vortices and other states in suspension Taylor-Couette flow, focusing on the dynamics and stability of these structures. A. Pinter *et al.* [8] investigate the influence of axial through-flow on the spatiotemporal growth behavior of different vortex structures in the Taylor-Couette system. F. S. Pereira *et al.* [9] use the partially averaged Navier-Stokes (PANS) equations to model transitional Taylor-Green vortex flow at Reynolds number 3000, focusing on vortex-stretching and reconnection mechanisms. On the other hand, Taylor vortices generated in the journal bearing can cause vibration and damage to the shaft, which can lead to machine failure. We came up with the idea of this research because we believe that if we can quickly and accurately predict the mode development process of the Taylor vortex, it will be possible to detect abnormal behavior and patterns, prevent shaft damage, and improve productivity.

In this study, images of numerically calculated Taylor vortices generated in a rotating double cylinder are merged horizontally for a certain period of time after vortex generation to confirm whether the mode development process can be judged accurately in the merged images. The objective is to construct a system that efficiently predicts the development process of vortices by combining images from a certain period of time after vortex generation under each analysis condition and training them to a convolutional neural network.

2. Analysis Procedure

2.1. Overview of Taylor Vortex Flow

The Taylor vortex flow is one of the most important eddies since the classical study. In a coaxially rotating double cylinder, when the circumferential speed of the inner cylinder is gradually increased from zero, a Couette flow appears between the inner and outer cylinders. Figure 1 shows a schematic diagram of the Taylor vortex flow. Taylor vortex flows can be broadly classified into two types: normal modes and mutant modes. When the top and bottom ends between the double cylinders are fixed, normal modes are those with an even number of vortices and flow from the outer cylinder to the inner cylinder at the top and bottom end faces, while other modes are called mutant modes. In the case of fixed ends, when an even number of cells are created with the bottom end clockwise and the top end counterclockwise, they are called normal cells, and all others are called mutated cells. As an example, Figure 2 shows a schematic diagram of two regular cells.

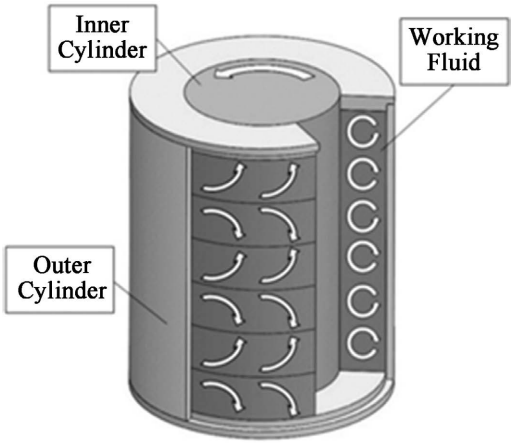

Figure 1. Taylor vortex flow.

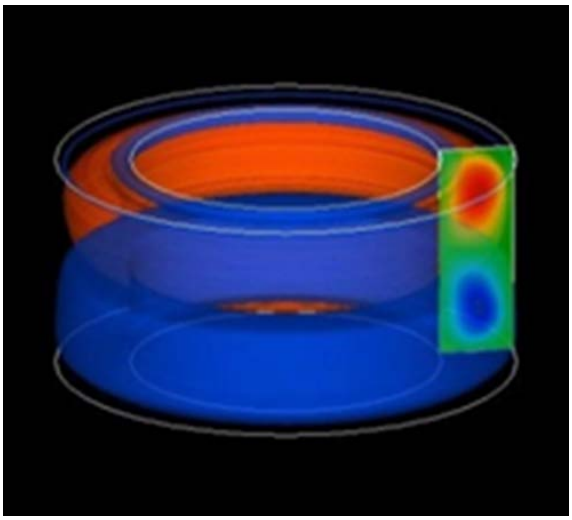

Figure 2. Normal cell.

2.2. Numerical Method

The governing equations are the axisymmetric unsteady incompressible Navier-Stokes equation with cylindrical coordinates (r, θ, z) (Equation (1)) and the equation of continuity (Equation (2)).

$$\frac{\partial u}{\partial t}+(u\cdot\nabla)u=-\nabla p+\frac{1}{Re}\nabla^2 u \tag{1}$$

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

Here, u is a velocity vector with the components of (u, v, w), t is a dimensionless time based on a representative time, and p is the pressure. The discretization of the governing equations is based on the MAC method. For time integration, Euler's explicit method is used. For space integration, the QUICK method is used for the convection terms, and the secondary central difference method is used for other terms. As an initial condition, the velocity is set to 0 in all regions. As boundary conditions, the no-slip condition at each cylinder wall is applied for the velocity, and the Neumann condition based on the equation of motion is applied for the pressure. Staggered grids are used as calculation grids and are assumed to have regular intervals in each direction. The number of the grid points in the radial direction is 41, and one in the axial direction is determined linearly according to the height of the cylinders.

2.3. Overview of Simulated Images

First, in order to collect images of the Taylor vortex flow, a computer simulation is performed and analyzed. The analysis method is as follows: the aspect ratio γ is set from 3.0 to 7.4, the Reynolds number Re is varied from 100 to 1000 in increments of 100, and the inner-cylinder acceleration time step number ac is analyzed from 0 to 10000 in increments of 1000 at each Reynolds number. The collected images were divided into a total of 10 folders, one for the normal mode (n2, n4, n6, n8, n10) and one for the mutated mode (a4, a5, a6, a7, a8) according to the number of vortices and direction of rotation, since labeling for each mode is necessary when training. Examples of simulated images of Taylor vortex flow in normal and mutant modes with respect to the flow function j are shown in Figure 3 and Figure 4,

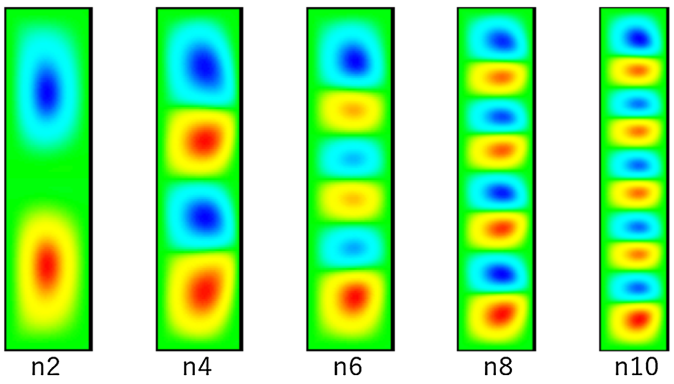

Figure 3. Normal mode (n2, n4, n6, n8, n10).

respectively. These images represent the final stages of the developmental process in each mode. As an example of the mode development process of the Taylor vortex, the normal two-cell development process of the flow function j (G = 3.1, Re = 100, ac = 2000) is shown in Figure 3. In this study, the images of the Taylor vortex development in each mode are horizontally combined for the images of the flow function j, and the images are used as teacher data for deep learning. The mode formation process of Taylor vortex flow is shown in Figure 5.

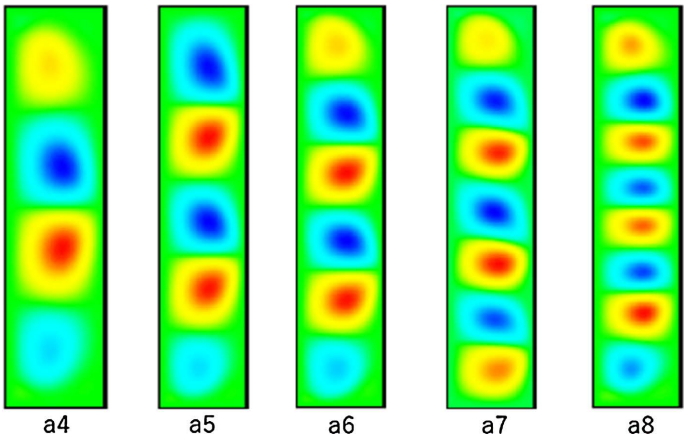

Figure 4. Anomalous mode (a4, a5, a6, a7, a8).

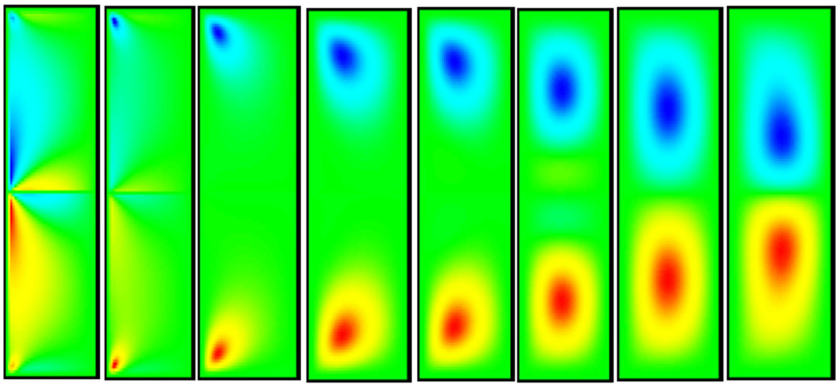

Figure 5. Taylor vortex development process.

2.4. Image Processing of Input Data

In this study, we first combine the images of the Taylor vortex flow simulation results with 200 images skipped in two from the vortex generation, and then train them into a convolution model to see if we can correctly determine the mode development process.

The images are combined and trained into convolutional models at 20 images at 1 second, 30 images at 1.5 seconds, 40 images at 2 seconds, 50 images at 2.5 seconds, 60 images at 3 seconds, and 67 images at 3.35 seconds, respectively, from vortex onset, and the accuracy rates in determining the mode development process are compared.

Initially, the images of Taylor vortices collected in the simulation are not all the

same size, so the images need to be merged horizontally after unifying the image sizes.

The average color of the RGB values is obtained for each of the images obtained from the simulation, and a square background image is created with the average color. After placing the simulated image on it, resize it.

Figure 6 shows a program to unify the image sizes. Figure 7 shows an example of a resized image. The resized images are then merged horizontally for an arbitrary time interval from vortex generation, and the resulting images are used as teacher data for deep learning. An example of a horizontally merged image (ini20) to be treated as teacher data is shown in Figure 8. The reason for combining images at specific time intervals is to capture patterns in time-series data. When using time series data, you can accurately understand changes in flow by considering time intervals. This allows the model to make predictions and classifications with higher accuracy.

```
def expand2square(img, h,w):

    width, height = img.size

    r_ave, g_ave, b_ave = 0, 0, 0
    for x in range(width):
        for y in range(height):
            r, g, b = img.getpixel((x,y))
            r_ave += r
            g_ave += g
            b_ave += b
    r = int(r_ave/(width*height))
    g = int(g_ave/(width*height))
    b = int(b_ave/(width*height))

    if width == height:
        new_img = img.resize((w,h))
        return new_img
    elif width > height:
        new_img = Image.new(img.mode, (width, width), (r,g,b))
        new_img.paste(img, (0, (width - height) // 2))
        new_img = new_img.resize((w,h))
        return new_img
    else:
        new_img = Image.new(img.mode, (height, height), (r,g,b))
        new_img.paste(img, ((height - width) // 2, 0))
        new_img = new_img.resize((w,h))
        return new_img
```

Figure 6. Unification of image size.

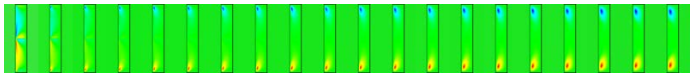

Figure 7. Resized image.

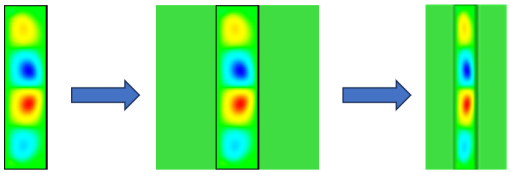

Figure 8. Concatenated image.

2.5. Convolutional Neural Networks

In this study, a convolutional neural network (CNN) is used to predict the mode development process of the Taylor vortex. First, convolutional neural networks are explained.

Convolutional neural networks are deep learning algorithms used primarily in the field of image recognition. It has a hierarchical structure with a convolutional layer that mainly extracts features and a pooling layer that reduces the resolution of the convolved data. In the convolutional and pooling layers, some regions of input neurons are narrowed down and locally mapped to the next layer. Each layer also has multiple detectors called kernels, with the first layer detecting edges and other information, and the deeper the layer, the more abstract features are detected. CNN automatically learns the parameters of the kernel, which is the detector for extracting these features. In the layer closest to the final output, classification, etc., is performed by all coupled layers.

Next, the processing of each layer is described. In the convolution layer, a weight matrix called a kernel slides over the image while performing convolution operations to extract different features and patterns. Figure 9 shows the processing in the convolution layer.

The operation is performed by sliding a weight vector, called a kernel, over the image data, which is the input data, at regular intervals. Specifically, the kernels are stacked sequentially starting from the upper left of the input data, the kernel elements are multiplied by the corresponding elements of the input data, and the sum of the kernels is obtained and stored in the corresponding location in the output result. The collection of values resulting from the convolution of all the elements of the input data with the kernel is called a feature map.

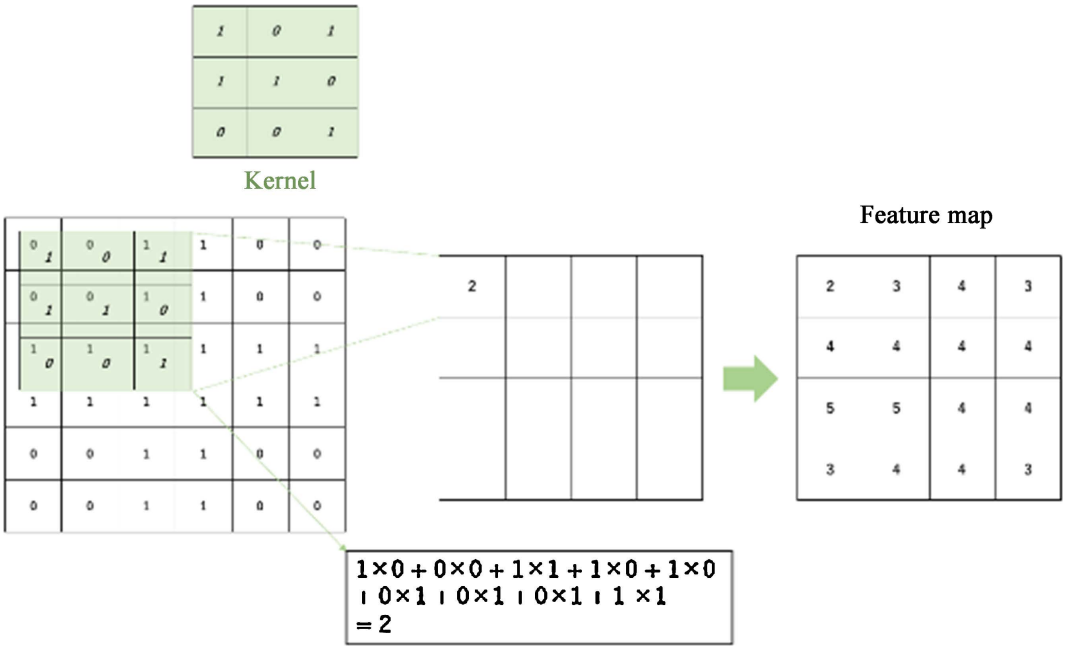

Figure 9. Convolutional layer.

The pooling layer aggregates the output of the convolutional layer to improve computational efficiency while increasing robustness to positional variations.

Figure 10 illustrates the process in the pooling layer. The purpose is to reduce the dimensionality of the data and lower the processing cost required for the computation. Pooling does not have the concept of learning, but only a fixed computation is performed. The activation function is a function used to convert the sum of input signals into an output signal.

Figure 11 shows the activation function. y_1. The function that processes the activation function is called the activation function $h(y_1)$ is called the activation function. The output is calculated by $y = h(y_1)$. The activation function is responsible for adjusting how the data is propagated to the next layer using the results of the calculations in the all-join and convolution layers as input.

Typical functions are listed below.

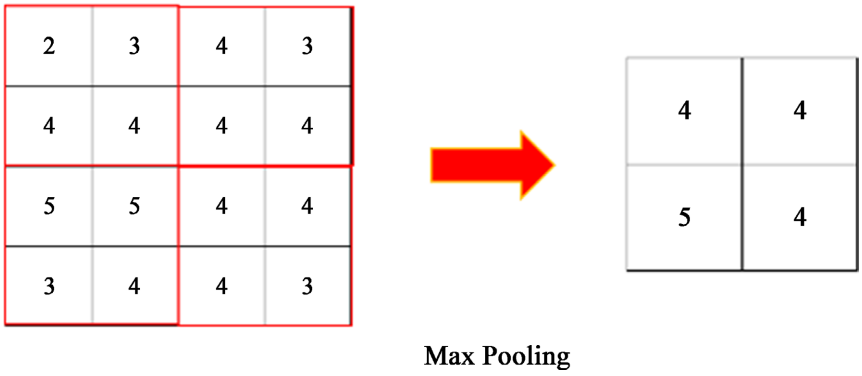

Figure 10. Pooling layer.

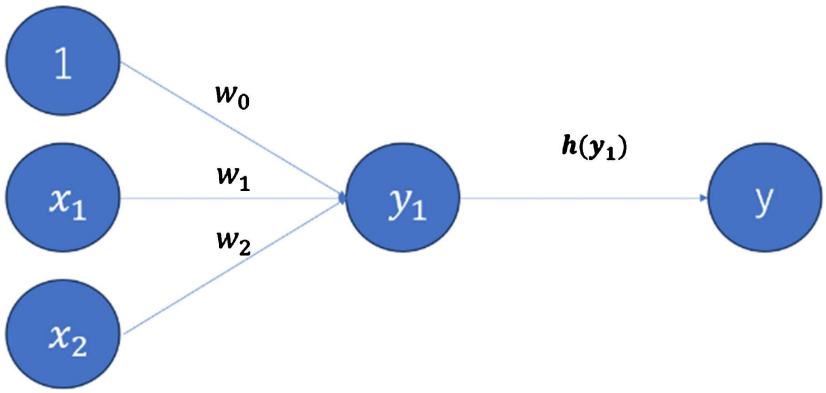

Figure 11. Activation function.

- ReLU (Rectified Linear Unit)

  ReLU is the activation function defined by the following equation

$$f(x) = \max(0, x)$$

Output value is zero if the input is less than or equal to zero, and the value is output as is if the input is greater than zero. Output as input value increases is also larger, so it learns faster.

- Sigmoid

Sigmoid is an activation function defined by the following equation.

$$f(x)=\frac{1}{1+e^{-x}} \tag{3}$$

This function converts the output value between 0 and 1.

- tanh

tanh (hyperbolic tangent function) is the activation function defined by the following equation.

$$f(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{4}$$

This function sets the output value between −1 and 1.

- softmax

Neural networks are mainly classified into classification and regression problems, and which classification determines the activation function of the final output layer. In general, softmax is used for classification problems and the identity function is used for regression problems. Softmax is the activation function defined by the following equation

$$f(x)=\frac{\exp(a_k)}{\sum\exp(a_i)} \tag{5}$$

It is a function whose individual outputs are 0 to 1 and the sum of the outputs is 1.0. This property allows us to treat the output of softmax as probability.

The all-junction layer is the most basic layer in a multilayer perceptron and is the layer responsible for feature-based classification. The respective features obtained by processing in the convolution and pooling layers are aggregated into a single node and output through the activation function. Convolutional neural networks automatically learn features from large amounts of data and perform well for complex pattern recognition tasks. They can also be applied to speech, text, and other data, and their versatility and feature learning capabilities have made them successful in a wide range of applications.

2.6. Flow of Mode Development Process Determination Using CNN

In this study, deep learning is performed to predict the mode development process of the Taylor vortex using simulated images coupled in the plane direction as teacher data.

The first step is to load images in order to train the model with the combined images as input data. In this study, a total of 10 horizontally combined images from folders a4, a5, a6, a7, a8, n2, n4, n6, n8, and n10 are read and used as input for the convolutional neural network. Figure 12 shows the process of reading an image, normalizing it to the range of 0 to 1, and labeling each mode.

Next, a convolutional neural network model is constructed. To develop a prediction system for the mode development process, horizontally combined images are read, and image features are extracted and condensed by convolution and

pooling layers.

The convolution and pooling layers output a two-dimensional array of outputs, but if the outputs remain two-dimensional, they cannot be input to all the coupling layers, so they are converted to a one-dimensional array in the flattening layer. The outputs obtained are then combined into a single node by all coupling layers, and the value transformed by the activation function is output.

Finally, based on the output from all the coupling layers, the modes are classified by converting them to probabilities using a softmax function and maximizing the probability of being correctly classified in each region. The model is then trained and stored. The model building program is shown in Figure 13. Draw a graph in matplotlib to check the loss and accuracy rate transitions during the learning process after the end of the training. Loss is the value of how much the actual correct value deviates from the prediction calculated by the learning model.

```
# データセットのルートディレクトリとクラス名
dataset_root = 'ini60'
class_names = ['a4','a5', 'a6', 'a7', 'a8', 'n2', 'n4', 'n6', 'n8', 'n10']

# データロードと前処理
X = []
Y = []
for index, class_name in enumerate(class_names):
    class_path = os.path.join(dataset_root, class_name)
    image_files = glob(os.path.join(class_path, "*"))
    for i, image_file in enumerate(image_files):
        image = Image.open(image_file)
        image = image.convert("RGB")
        data = np.asarray(image) / 255.0
        X.append(data)
        Y.append(index)
        print(class_name, i, len(image_files))

X = np.array(X)
Y = np.array(Y)
Y = to_categorical(Y, len(class_names))

# データ分割
X_train, X_val, y_train, y_val = train_test_split(X, Y, test_size=0.20)
```

Figure 12. Preparation of learning data.

```
# モデルの定義
model = Sequential()
model.add(Conv2D(32, (3, 3), activation="relu", input_shape=(image_height, image_width, num_channels)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation="relu"))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(500, activation='relu'))
model.add(Dense(200, activation='relu'))
model.add(Dense(50, activation='relu'))
model.add(Dense(len(class_names), activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='SGD', metrics=['accuracy'])

# モデルの訓練
history = model.fit(X_train, y_train, batch_size=batch_size, epochs=epochs, validation_data=(X_val, y_val))

# モデルの保存
model.save("result/taylor" + str(datetime.date.today()) + ".h5", include_optimizer=False)
```

Figure 13. Building a machine learning model.

Since learning models do not necessarily compute accurate data as predictions, a loss function is used to compute the loss. In the task of multi-class classification in deep learning, cross-entropy error is generally used for the loss function, and cross-entropy error is also used in this study. The graphs of loss and accuracy rate are paired, and the goal is to achieve a learning result where the value of loss is small and the value of accuracy rate is large. A program that draws a graph of loss and accuracy rates during the learning process is shown in Figure 14. Figure 15 shows an example graph of loss and accuracy during the learning process. In both graphs, the horizontal axis is the number of epochs, which is the number of iterations of learning.

```
# 学習履歴のプロット
fig, (axL, axR) = plt.subplots(ncols=2, figsize=(10, 4))

def plot_history_loss(fit):
    axL.plot(fit.history['loss'], label="loss for training")
    axL.plot(fit.history['val_loss'], label="loss for validation")
    axL.set_title('model loss')
    axL.set_xlabel('epoch')
    axL.set_ylabel('loss')
    axL.legend(loc='upper right')

def plot_history_acc(fit):
    axR.plot(fit.history['accuracy'], label="acc for training")
    axR.plot(fit.history['val_accuracy'], label="acc for validation")
    axR.set_title('model accuracy')
    axR.set_xlabel('epoch')
    axR.set_ylabel('accuracy')
    axR.legend(loc='lower right')

plot_history_loss(history)
plot_history_acc(history)
fig.savefig('result/Taylor.png')
plt.show()
```

Figure 14. Display of learning results.

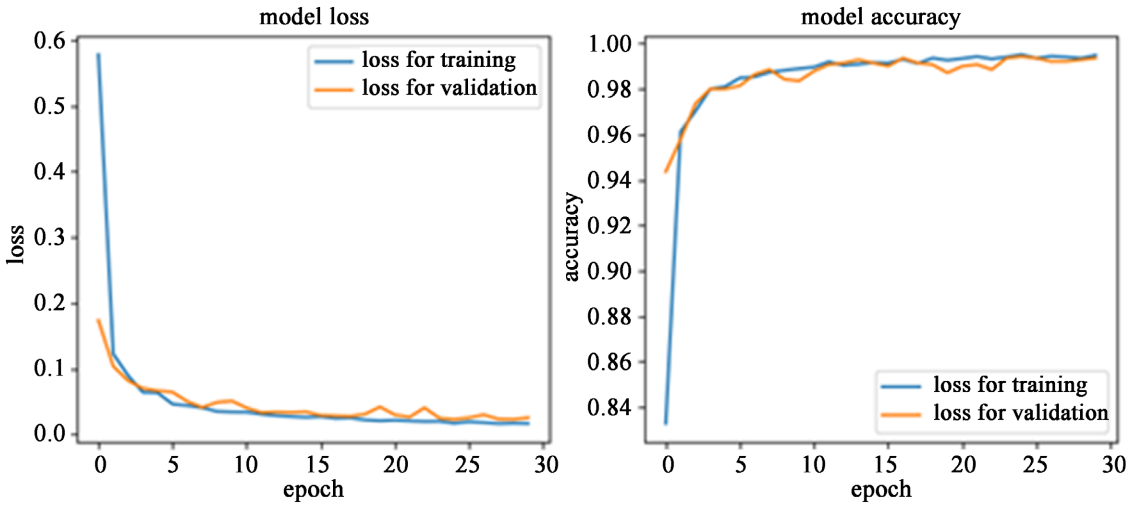

Figure 15. Loss and accuracy.

Finally, we verify whether the predictions calculated by the learning model are consistent with the actual correct values. In this study, images are randomly selected from the teacher data and grouped together in a test folder to check whether the prediction of the mode development process is correct. After the test images are loaded, predictions are made using the learned model. Figure 16 shows a program for loading test images from the test folder and making predictions. Figure 17 shows the program to verify the test image, and Figure 18 shows the resulting output screen.

In this study, 200 images are acquired in duplicate from the vortex onset and combined to train the model to confirm whether the mode development process is correctly predicted. In addition, 20 images at 1 second, 30 images at 1.5 seconds, 40 images at 2 seconds, 50 images at 2.5 seconds, 60 images at 3 seconds, and 67 images at 3.35 seconds from vortex onset were each combined horizontally and trained into the model to compare the accuracy rate of the prediction results.

```
# モデルの評価
test_folder = "test"
test_dir = 'ini60/'

test = []
files = glob(os.path.join(test_dir, test_folder, '*'))
for file in files:
    image = Image.open(file)
    image = image.convert("RGB")
    data = np.asarray(image) / 255.0
    test.append(data)

test = np.array(test)

result = model.predict(test)
```

Figure 16. Predicting mode.

```
# モデルによるテストデータ判定結果の取得
determined_mode = []
for i in result:
    mode = str(class_names[np.argmax(i)])
    determined_mode.append(mode)

# 正解率の判定
count = 0
for i in range(len(files)):
    if (test_files_mode[i] == determined_mode[i]):
        count+= 1
        acc_sym = "o"
    else:
        acc_sym = "×"

    print(test_files_mode[i], determined_mode[i], acc_sym)

print(f"テスト数：{len(files)}      正解数：{count}")

print(f"正解率：{float(count/len(files))*100}%")
```

Figure 17. Display prediction results.

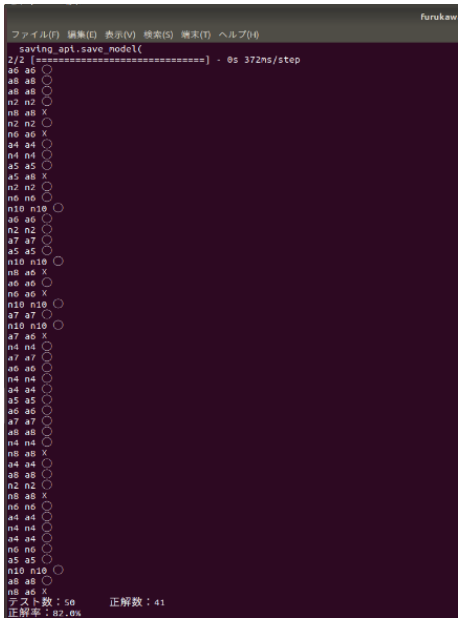

Figure 18. Result of mode prediction.

There are several factors that affect the predictive accuracy of a model during its evolution. High-resolution, clear images help models extract features more accurately and improve prediction accuracy. Conversely, low-quality images can be noisy and reduce prediction accuracy. Also, the larger the dataset, the more patterns the model learns and the better its generalization ability. Small datasets increase the risk of overfitting, a phenomenon in which a model overfits the training data and performs poorly on new data.

3. Results

Figure 19 shows the accuracy rate and loss when 200 images (10 seconds) were acquired after vortex generation, combined, and trained as supervised data.

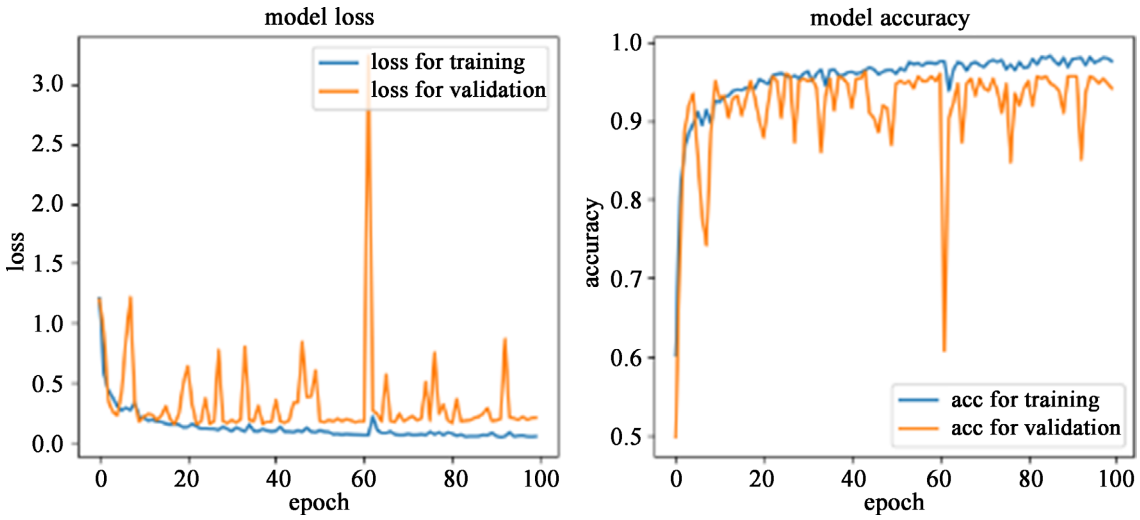

Figure 19. Changes in loss and accuracy rate.

Fifty random merged images were loaded into the model created as test images, and the results are shown in Figure 20 to verify whether mode judgments could be made. It shows that the mode development process can be accurately predicted. Next, 20, 30, 40, 50, 60, and 67 images from the vortex generation were combined and used as teacher data to create a model, and the test results are shown in Table 1.

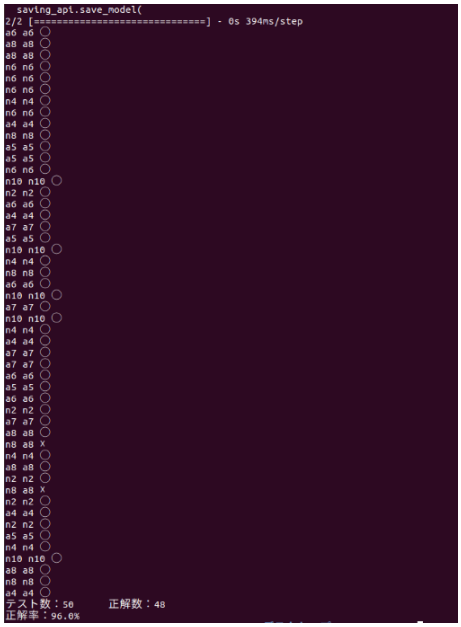

Figure 20. Result of mode prediction on test.

Table 1. Prediction results.

Image Num.	ini20	ini30	ini40	ini50	ini60	ini67
Accuracy	68%	82%	90%	94%	94%	98%

4. Discussion

In this study, we confirm that the mode determination of vortices by convolutional neural networks using the combined images is correct, compare the accuracy rate while changing the time range of the combined images, and evaluate the appropriate time range in which the prediction of the mode development process can be said to be possible. As shown in Table 1, it can be seen that the accuracy rate of mode development process determination increases as the number of images to be combined increases, *i.e.*, as the Taylor vortex between rotating double cylinders undergoes mode development. Among them, the accuracy rate in the mode development process judgment was high when 50 images during 2.5 seconds from the vortex onset were trained to the convolutional neural network as teacher data, and it can be seen that the accuracy rate did not change when the number of images was increased to 60. Therefore, in this study, we believe that the mode development process of the Taylor vortex between rotating double cylinders

can be efficiently predicted by having the convolutional neural network learn 50 images for 2.5 seconds after the Taylor vortex is generated.

5. Conclusions

In this study, the numerical results of Taylor vortex flow between rotating double cylinders were combined for a certain period of time after vortex onset using a convolutional neural network, and it was shown that mode discrimination was possible in the combined images. In addition, 20 images at 1 second, 30 images at 1.5 seconds, 40 images at 2 seconds, 50 images at 2.5 seconds, 60 images at 3 seconds, and 67 images at 3.35 seconds after vortex onset were combined and used as input images to the convolutional neural network to compare the accuracy rate in determining the mode development process of the vortex. The following are the conclusions obtained from this study.

It was confirmed that the mode development process of the Taylor vortex can be discriminated by combining images of a certain period of time after the occurrence of the Taylor vortex flow and training the convolutional neural network with those images as teacher data.

The results of image recognition using the images merged horizontally for 20 images at 1 second, 30 images at 1.5 seconds, 40 images at 2 seconds, 50 images at 2.5 seconds, 60 images at 3 seconds, and 67 images at 3.35 seconds after vortex onset, respectively, showed that learning with 50 images at 2.5 seconds was the most efficient way to predict the mode development process. The results showed that the most efficient prediction of the mode development process was achieved when 50 images of 2.5 seconds were used for learning.

Convolutional neural networks (CNNs) are a very promising technique in fluid mechanics research. As for application prospects, CNN will also be utilized as a tool to predict future hydrodynamic phenomena. For example, it could be useful in predicting the effects of floods and typhoons. As an industrial and scientific value, research on fluid dynamics using CNN contributes to improving energy efficiency. For example, optimizing the aerodynamic design of aircraft and cars can reduce fuel consumption. CNN has great potential in these areas. This study made a major contribution to increasing these possibilities.

Acknowledgements

We would like to show our gratitude to members of our laboratory for sharing their pearls of wisdom with us during the course of this research.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Krueger, E.R., Gross, A. and Prima, R.C.D. (1966) On the Relative Importance of Taylor-Vortex and Non-Axisymmetric Modes in Flow between Rotating Cylinders.

Journal of Fluid Mechanics, **24**, 521-538. https://doi.org/10.1017/s002211206600079x

[2] Jones, C.A. (1981) Nonlinear Taylor Vortices and Their Stability. *Journal of Fluid Mechanics*, **102**, 249-261. https://doi.org/10.1017/s0022112081002620

[3] Edwards, M.D., Pratley, M.T., Gordon, C.M., Teixeira, R.I., Ali, H., Mahmood, I., *et al*. (2024) Process Intensification of the Continuous Synthesis of Bio-Derived Monomers for Sustainable Coatings Using a Taylor Vortex Flow Reactor. *Organic Process Research & Development*, **28**, 1917-1928. https://doi.org/10.1021/acs.oprd.3c00462

[4] Wang, Z., Lyu, Z. and Jiang, J. (2024) Induction of Controllable Vortical Flow in a Dual-Stenosis Aorta Model: A Replication of Disordered Eddies Flow in Aneurysms. *Journal of Cardiovascular Translational Research*. https://doi.org/10.1007/s12265-024-10566-y

[5] Liu, D., Wang, Y., Shi, W., Kim, H. and Tang, A. (2015) Slit Wall and Heat Transfer Effect on the Taylor Vortex Flow. *Energies*, **8**, 1958-1974. https://doi.org/10.3390/en8031958

[6] Yang, Z., Deng, B., Wang, B. and Shen, L. (2020) Sustaining Mechanism of Taylor-Görtler-Like Vortices in a Streamwise-Rotating Channel Flow. *Physical Review Fluids*, **5**, Article ID: 044601. https://doi.org/10.1103/physrevfluids.5.044601

[7] Ramesh, P. and Alam, M. (2020) Interpenetrating Spiral Vortices and Other Coexisting States in Suspension Taylor-Couette Flow. *Physical Review Fluids*, **5**, Article ID: 042301. https://doi.org/10.1103/physrevfluids.5.042301

[8] Pinter, A., Lücke, M. and Hoffmann, C. (2003) Spiral and Taylor Vortex Fronts and Pulses in Axial through Flow. *Physical Review E*, **67**, Article ID: 026318. https://doi.org/10.1103/physreve.67.026318

[9] Pereira, F.S., Grinstein, F.F., Israel, D.M., Rauenzahn, R. and Girimaji, S.S. (2021) Modeling and Simulation of Transitional Taylor-Green Vortex Flow with Partially Averaged Navier-Stokes Equations. *Physical Review Fluids*, **6**, Article ID: 054611. https://doi.org/10.1103/physrevfluids.6.054611