International Journal of Scientific Research & Engineering Trends
Volume 9, Issue 3, May-Jun-2023, ISSN (Online): 2395-566X

Study on Determining Taylor Vortex Flow Mode Development Process Using Various Physical Quantities

Hiroyuki Furukawa, Takeomi Yamazaki
Department of Mechanical Engineering
Meijo University
Nagoya, Japan
furukawa@meijo-u.ac.jp

Abstract- In recent years, technology that harnesses the unlimited potential of microorganisms has become important as a modest but long-lasting technology. In order to maximize the power of microorganisms, it is necessary to control the flow of culture medium to mix them uniformly, light, carbon dioxide. Taylor vortices are considered suitable for agitated culture of plant and animal cells or microorganisms because they are easy to create and are resistant to disturbances, stable, and have little local shear flow. In this study, we constructed a system that can automatically discriminate the flow mode using numerical results of Taylor vortex flow generated between rotating double cylinders as input data by using deep learning. By comparing the loss and accuracy rate of test data for various physical quantities and comparing the accuracy rate and loss of training data, the physical quantities that can efficiently predict the mode development process of the Taylor vortex were shown. The results show that among the various physical quantities, the radius u is the most accurate when comparing the final loss after learning and the accuracy rate and can efficiently predict mode development process of the Taylor vortex..

Keywords- Taylor vortex flow, Deep learning, Flow Visualization method, CFD.

I. INTRODUCTION

In recent years, technologies that harness the unlimited potential of microorganisms have become increasingly important, and though inconspicuous, they are considered to be long-lasting technologies. For example, Denso Corporation is conducting research at a large-scale microalgae culture demonstration facility in Amakusa City in Kumamoto Prefecture, with the aim of establishing a technology to utilize fuel extracted from Pseudochoricystis ellipsoidea, an algae that can produce oil, for automobiles. Oxygen is essential for many organisms on the earth, and the amount of oxygen emitted by microalgae through photosynthesis is almost the same as the amount of oxygen produced by terrestrial plants.

Therefore, microalgae support the environment of the Earth as photosynthetic organisms that convert carbon dioxide into oxygen. Additionally, research is being conducted on a system (bioreactor) in which a large amount of oxygen is produced by microorganisms in microalgae that have a photosynthetic capacity 30 to 50 times greater than that of plants. To maximize the power of microorganisms, it is necessary to control the flow of light, carbon dioxide and the culture medium so that they can be mixed uniformly. As cells of microorganisms are damaged by strong agitation, a Taylor vortex [1] with stable but gentle agitation is used. Since Taylor vortices are easy to create, stable resistant to disturbances, and have little local shear flow, they are considered suitable for the agitation and culture of plant and animal cells or microorganisms [2].

The present study was inspired by the possibility of efficiently predicting the mode development process of the Taylor vortex so as to control the flow when stirring and uniformly mixing microorganisms.

The rapid development of artificial intelligence in recent years has led to its practical application in a wide variety of fields. The purpose of this study is to construct a system that can use Deep Learning to automatically determine flow modes and evaluate various physical quantities using the numerical results of a Taylor vortex flow between rotating double cylinders as input data. The flow function psi, pressure p, radius u, circumference v, and axis w are used as the various physical quantities.

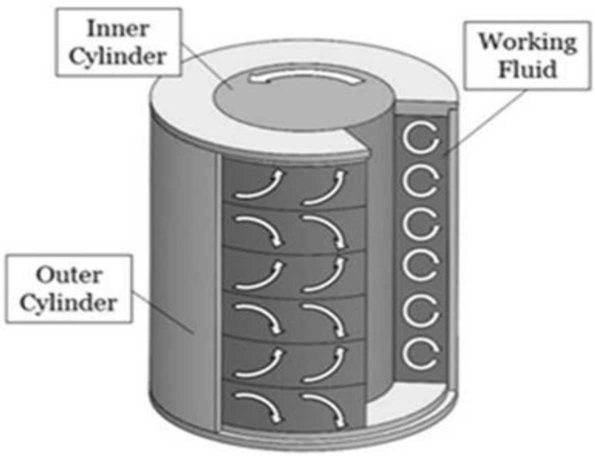


Fig 1. Taylor vortex flow.

International Journal of Scientific Research & Engineering Trends
Volume 9, Issue 3, May-Jun-2023, ISSN (Online): 2395-566X

II. ANALYSIS METHODS

Since the classical study by G.I.Taylor in 1923, the Taylor vortex flow has been studied as one of the most important vortices. In a coaxially rotating double cylinder, a Couette flow appears between the inner and outer cylinders as the circumferential velocity of the inner cylinder is gradually increased from zero. When the circumferential velocity is further accelerated, it has been confirmed that there is a mode transition to a Taylor vortex flow, which consists of multiple layers of flow on a torus called a cell, and then to a wave Taylor vortex flow. An overview of the Taylor vortex flow is shown in Fig.1.

The Taylor vortex flow can be broadly classified into two types: normal mode and anomalous mode. When the upper and lower ends between the two cylinders are fixed, the normal mode is the mode in which the flow progresses from the outer cylinder to the inner cylinder at the upper and lower ends, and the anomalous mode is the mode in which the flow progresses from the inner cylinder to the outer cylinder at the upper and lower ends or at one of the end faces. In addition, in the case of fixed ends, when an even number of cells are formed with the lower end clockwise and the upper end counterclockwise, they are called normal cells, and all other cells are called mutant cells. As an example, Fig. 2 shows a schematic diagram of two normal cells.

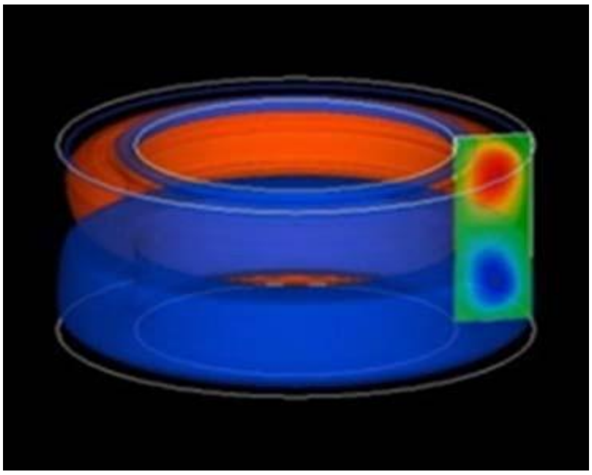


Fig 2. Normal cell.

In this study, we use images of Taylor vortex flow simulation results as the training data, and use a method called supervised learning to teach the machine. First, in order to collect images of the Taylor vortex flow simulation results to serve as the training data, we perform the simulation on a computer using Linux and analyze the data. 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 ac is analyzed from 0 to 10000 seconds in increments of 1000 for each Reynolds number. As it is necessary to distinguish between the collected images based on the learned conditions when learning, the collected images are divided into a total of 10 folders of normal and mutated modes under conditions in which the images can be distinguished by the number of vortices.

Machine learning is performed for five patterns of flow function psi, pressure p, radius u, circumference v, and axis w, and they are divided respectively into 10 folders. As examples of the images of the simulation results for the Taylor vortex flow in the normal and mutant modes, images for the flow function psi are shown in Fig.3 and Fig.4, respectively. These images represent the final developmental stage in each mode. As an example of the mode development process of the Taylor vortex, the development process of two normal cells with the flow function psi (γ=3.1, Re=100, ac=2000) is shown in Fig.5. In Fig.5, one cell is generated at each of the top and bottom in the initial stage, and the cells develop from the top and bottom edges toward the center, eventually resulting in the generation of a total of two cells, which are n2 cells.

In this study, we use Deep Learning to predict the mode of development of the Taylor vortex using the input data (training data). First, we explain the flow of Deep Learning. In order to predict the mode, it is necessary to first input training data for the machine to learn. After reading the input data, Deep Learning is performed. Lastly, if the learning results correct or not is verified. Fig.6 shows the flow of Deep Learning using the input data.

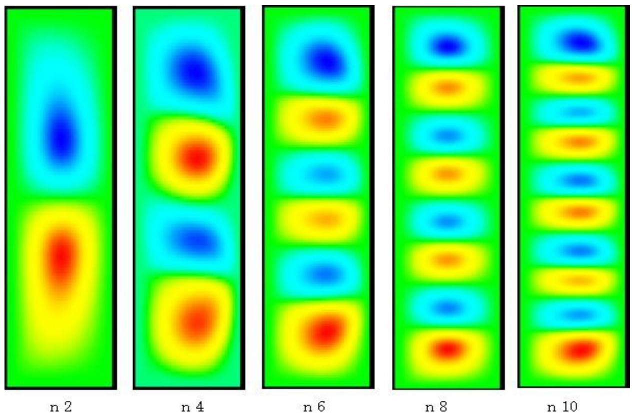


Fig 3. Normal mode(n2,n4,n6,n8,n10).

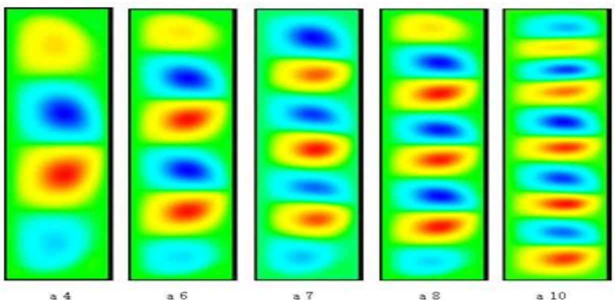


Fig 4. Anomalous mode(a4,a6,a7,a8,a10).

International Journal of Scientific Research & Engineering Trends
Volume 9, Issue 3, May-Jun-2023, ISSN (Online): 2395-566X

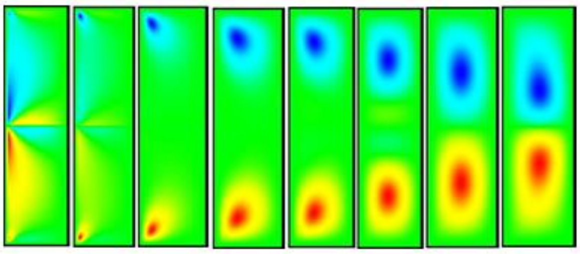


Fig 5. Taylor Vortex Development Process

Next, it is necessary to unify the size of the images used as the input data because the images of the training data collected in this study are not all of the same size. Resizing them enables the machine to learn them. Since the images used in this study are vertically (2:1), they are placed on top of a square image and the areas that are not required are filled in to make the image size in Fig 7 uniform. Fig 8 shows an example of an image that has been resized.

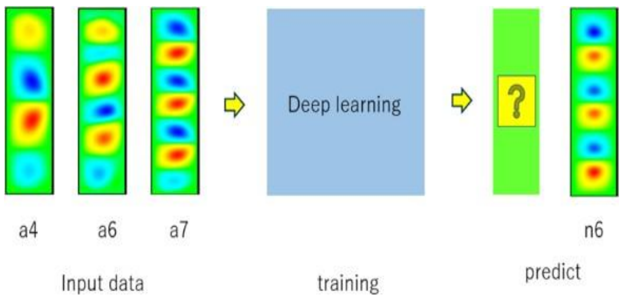


Fig 6. Deep learning using input data.

```python
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
```

Fig 7. Unification of image size.

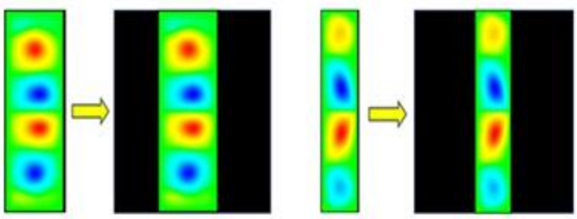


Fig 8. Resized image.

Next, we load the input data into the machine so that the machine can learn the images. Since the machine is unable to process all the collected images due to the large quantity of data, we perform Deep Learning by loading 10,000 images each from 10 folders from a4 to n10, making a total of 100,000 images in this study. Fig 9 shows the program used for loading the images. In Fig 9, images a4 to n10 are labeled 0 to 9. X=[ ] are the images in the training data and Y=[ ] are labels 0 to 9 corresponding to these images.

Next, we shall determine the number of epochs to be trained by the machine in this study. The number of iterations of training a single training data is called the number of epochs, and in general, the accuracy rate approaches 1.00 as the number of epochs increases.

Learning that does not work well for samples other than the training data because it tries to train the learning to fit the training data perfectly is called over-learning [3]. When over-learning occurs, the accuracy of the test data does not improve even though the accuracy of the training data improves as the number of epochs increases. Furthermore, the data used to train the machine is training data, and the data used to check how well the machine is predicting is test data. In this study, the accuracy rate is checked in the training process, and the machine is trained with an epoch number of 100 when the accuracy rate begins to converge. Fig 10 shows the program that specifies the number of epochs and performs learning.

```python
folder = ['a4','a6','a7','a8','a10','n2','n4','n6','n8','n10']
X = []
Y = []
for index, name in tqdm.tqdm(enumerate(folder)):

    files = glob.glob('Taylor_file_psi/'+name + "/*/*")
    for i, file in enumerate(files):
        image = Image.open(file)
        image = image.convert("RGB")
        image = expand2square(image,80,40)
        data = np.asarray(image)
        X.append(data)
        Y.append(index)
        print(name, i, len(files))

X = np.array(X)
Y = np.array(Y)

X = X.astype('float32')
X = X / 255.0

# 正解ラベルの形式を変換
Y = np_utils.to_categorical(Y,len(folder))

# 学習用データとテストデータ
X_train, X_val, y_train, y_val = train_test_split(X, Y, test_size=0.20)
```

Fig 9. Preparation of learning data.

```python
history = model.fit(X_train, y_train,
                    epochs=100,
                    validation_data=(X_val,y_val))

model.save("result/taylor"+str(datetime.date.today())+".h5",include_optimizer=False)
```

Fig 10. Epoch program.

International Journal of Scientific Research & Engineering Trends
Volume 9, Issue 3, May-Jun-2023, ISSN (Online): 2395-566X

```
fig, (axL, axR) = plt.subplots(ncols=2, figsize=(10,4))

# loss
def plot_history_loss(fit):
    # Plot the loss in the history
    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')

# acc
def plot_history_acc(fit):
    # Plot the loss in the history
    axR.plot(fit.history['accuracy'],label="loss for training")
    axR.plot(fit.history['val_accuracy'],label="loss 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/Tylor.png')
plt.show()
```

Fig 11. Display of learning results.

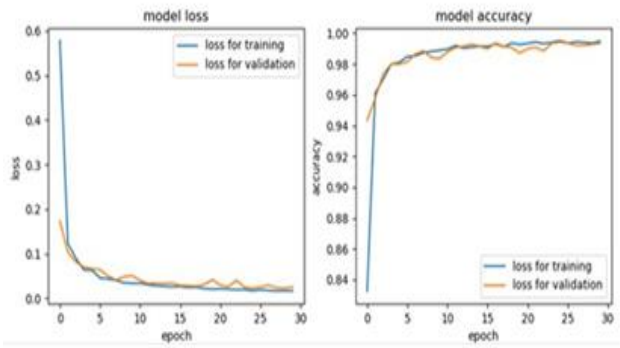


Fig 12. Loss function and accuracy rate.

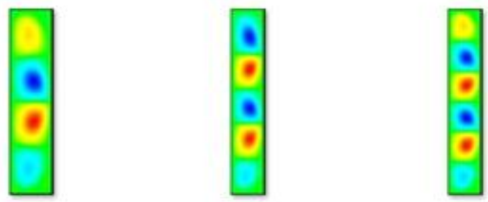
After the learning is completed, the loss function and the accuracy rate of the learning are checked. The loss function is a function that evaluates the degree of error between the output of the model and the training data during learning. The graphs of the loss function and the accuracy rate are paired, and the goal is to obtain learning results with a low loss and high accuracy rate. A program that displays the loss function and accuracy rate is shown in Fig 11. Fig 12 shows examples of the loss function and accuracy rate after the end of learning. In Fig 12, the vertical axes are indicated by loss for loss and accuracy for accuracy, and both of the horizontal axes indicate the number of epochs. Loss for training is the graph showing the learning data and loss for validation is the graph of the test data.

Lastly, we verify whether the learning results are correct or not. In this study, we randomly select images from the training data and compile them into a test folder to check whether the mode of those images has been correctly determined. Fig 13 shows the program used for verification. Fig 14 shows the randomly selected images and Fig 15 shows the verification results. In Fig 15, the result of a4 is indicated as "a4" in the first line, a5 as "a5" in the second line, a6 as "a6" in the third line, and so on. Compared with Fig 14, it shows that the developmental process of the mode is correctly determined. This is the Deep Learning flow. In this study, we compare the results of machine learning for five patterns of flow function psi, pressure p, radius u, circumference v, and axis w of various physical quantities.

```
test=[]
test_falder = "test"
dirc = 'Taylor_file_psi/'

files = glob.glob(dirc +'/'+test_falder+'/*')
for file in files:
    image = Image.open(file)
    image = image.convert("RGB")
    image = expand2square(image,80,40)
    data = np.asarray(image)
    test.append(data)
test = np.array(test)
test = test.astype('float32')
test= test / 255.0
```

Fig 13. Program for predicting mode.



a4_0079.gif a5_0807.gif a6_0842.gif

Fig 14. Test image.

```
['Taylor_file//test¥¥a4_0079.gif', 'Taylor_file//test¥¥a5_0807.gif', 'Taylor_file//test¥¥a6_0842.gif']
a4
a5
a6
```

Fig 15. Mode prediction.

III. ANALYSIS RESULTS

The results of machine learning of five patterns, namely flow function psi, pressure p, radius u, circumference v, and axis w, are shown in Fig 16 to Fig 20.

We compared the final loss and accuracy rates of various physical quantities in regard to the flow function psi, pressure p, radius u, circumference v, and axis w used in this study, and confirmed how accurately each pattern determines the mode development process. Table 1 shows the loss and accuracy rates of the five patterns of flow function psi, pressure p, radius u, circumference v, and axis w when loss for training (learning data) and loss for validation (test data) are at epoch 100. The numerical values of loss, accuracy, val_loss, and val_accuracy in Table 1 represent the final loss and accuracy rates of Fig 16 to Fig

International Journal of Scientific Research & Engineering Trends
Volume 9, Issue 3, May-Jun-2023, ISSN (Online): 2395-566X

20, respectively. In Table 1, loss represents the loss of learning data, accuracy represents the accuracy rate of learning data, val_loss represents the loss of test data, and val_accuracy represents the accuracy rate of test data.

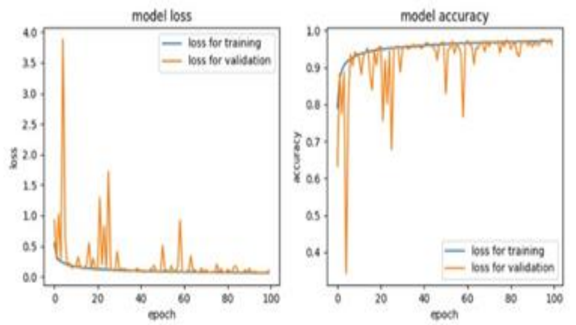


Fig 16. Loss function and accuracy rate (psi).

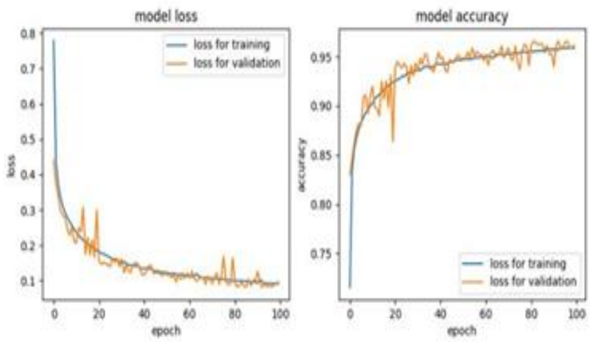


Fig 17. Loss function and accuracy rate (p).

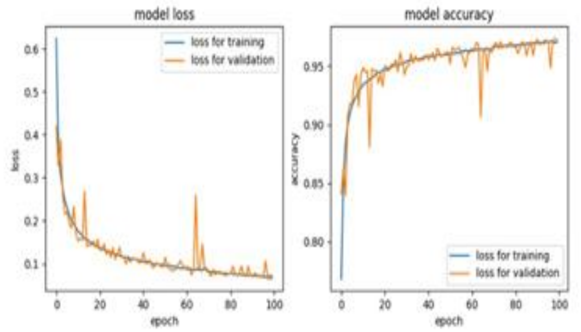


Fig 18. Loss function and accuracy rate (u).

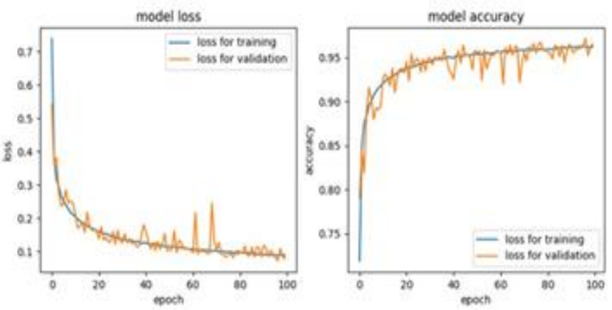


Fig 19. Loss function and accuracy rate (v).

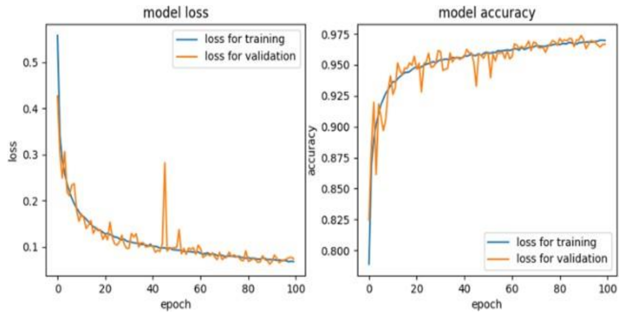


Fig 20. Loss function and accuracy rate (w).

Table 1. Font Sizes for Papers.

Epoch=100				
	loss	accuracy	val_loss	val_accuracy
psi	0.0633	0.9721	0.1061	0.9606
p	0.0918	0.9596	0.1137	0.9552
u	0.0700	0.9705	0.0665	0.9718
v	0.0867	0.9625	0.0770	0.9649
w	0.0678	0.9698	0.0734	0.9667

First, we compared the loss and accuracy rates of the test data of the various physical quantities. From Table 1, we can identify that the values of val_loss are smaller in the order of pressure p, flow function psi, circumference v, axis w, and radius u, while the values of val_accuracy are larger in the order of pressure p, flow function psi, circumference v, axis w, and radius u. The values of val_accuracy increase in the order of pressure p, flow function psi, circumference v, axis w and radius u. These results indicate that the radius u has the highest accuracy in determining the mode development process, since the accuracy of the learning results is higher when the loss is small and the accuracy rate is high. Next, we compare the loss and val_loss, accuracy and val_accuracy of various physical quantities.

For radius u and circumference v, loss > val_loss and accuracy < val_accuracy, indicating that the test data has a higher accuracy rate and less loss than the learning data, and thus can determine the mode development process efficiently. Conversely, the flow function psi, pressure p, and axis w are loss < val_loss , accuracy > val_accuracy, indicating that the test data has a lower accuracy rate and more loss than the learning data, and therefore the mode development process could not be determined as efficiently as the radius u and circumference v. This indicates that the test data could not determine the mode development process as efficiently as the radius u and circumference v. From these results, we infer that the radius u is the most accurate among the various physical quantities in this study and may be able to efficiently predict the mode development process of the Taylor vortex.

IV. CONCLUSION

In this study, we constructed a system that can automatically determine the flow mode using the numerical results of Taylor vortex flow between rotating double cylinders as input data with Deep Learning, and compared various physical quantities, namely flow function psi, pressure p, radius u, circumference v, and axis w.

The conclusions obtained from this study are as follows. It is now possible to automatically determine the mode development process of the Taylor vortex flow from a4 to n10 for various physical quantities. Comparing the final loss and accuracy rates after learning the various physical quantities, it was discovered that the radius u is the most efficient in determining the mode development process.

REFERENCES

[1] Benjamin, T. B., "Bifurcation Phenomena in Steady Flows of a Viscous Fluid. II. Experiment", Proc. R. Soc. London, Ser. A, 359, pp. 27-43, 1978.
[2] Tatsuya Fujii et al, "Frequency Analysis of Chaotic Flow in Transition to Turbulence in Taylor-Couette System with Small Aspect Ratio", Journal of Physics: Conference Series, 801 012063, 2016.
[3] Francois Chollet, "Deep Learning with Python", Simon and Schuster, 2021.