Convolutional Neural Network-Based Automatic Classification for Algal Morphogenesis

Kohma Hayashi, Shoichi Kato and Sachihiro Matsunaga*

Department of Applied Biological Science, Faculty of Science and Technology, Tokyo University of Science, 2641 Yamazaki, Noda, Chiba 278–8510, Japan

Received March 8, 2018; accepted April 19, 2018

Summary Convolutional neural networks (CNNs) are used for various data analyses and resemble the human brain cognition system. CNN algorithms are composed of multiple layers of convolution and pooling layers. Recent studies have shown that applying CNN algorithms to classify biological images are feasible. The alga, which has various morphological features, is the one of the interesting targets to classify using CNN. As an example, we targeted unicellular Cyanidioschyzon merolae. C. merolae is a primitive red alga that is not only used for studying organelles but also to proceed the study of the production of lipids. Measuring the division rate requires the classification of the interphase and mitotic phase of C. merolae. In this study, we constructed an automatic classifier for interphase and mitotic C. merolae. By tuning the hyper-parameter, number of layers, and number of insertions of dropout functions, we classified the interphase and mitotic C. merolae with the high accuracy of 92%.

Key words Cyanidioschyzon merolae, Convolutional neural network, Auto classification, Deep learning, Mitotic phase.

Convolutional neural networks (CNNs) have attracted attention as an approach to classifying biological images (Kamatani et al. 2017, Trebeschi et al. 2017, Xu et al. 2017, Niioka et al. 2018). Some automatic classification tools for biological images, such as CellProfiler (Carpenter et al. 2006), CellTrack (Sacan et al. 2008), CL-Quant (Lin et al. 2010), and CARTA (Kutsuna et al. 2012), have to extract image features. A CNN is a type of artificial neural network (ANN). ANNs are multilayered neural networks that resemble the human brain cognition system. The first model to apply the neural system was called MCP model (McCulloch and Pitts 1943). After a decade, the concept of the perceptron, which was based on the MCP model and the simplest form of the neural network, was introduced (Rosenblatt 1957). The simple perceptron is composed of two layers, an input layer, and an output layer. The output data is classified by activation of functions such as unit step function and sigmoid function. Although ANNs were introduced in 1957, they have encountered several problems, such as a lack of computing power, the disability to calculate the exclusive disjunction, and the difficulty in applying to the multilayer model with more than 4 layers. These problems have been solved by recent progress in computing science, for example, the appearance of the graphics processing unit (reviewed in Lee et al. 2017), and learning algorithms, such as backpropagation (Rumelhart et al. 1986) and autoencoder (Hinton and Salakhutdinov 2006). After CNN was applied to the recognition of a document (LeCun et al. 1998), it was applied to various data processing approaches, such as natural language processing (Collobert and Weston 2008), content-based music recommendation (den Oord et al. 2013), and bioinformatics (Min et al. 2017).

CNN is composed of multiple layers of convolution and pooling layers. In a CNN, characteristics are extracted by convoluting the input images. Images are convoluted to smaller images, and each part of the image is used as an indicator of the classification. These convoluted images compose the convolution layers. To process the image easily, the convoluted images are pooled. The pooled images compose the pooling layers. In a CNN, learning is executed by repeating the convolution and pooling the images.

C. merolae is a unicellular red alga, which possesses a minimum set of organelles, sole nucleus, sole mitochondrion, and sole chloroplast. This simple structure makes the organelles straightforward to study, and many dynamic phenomena have been determined, such as the distribution of organelles (Kuroiwa et al. 2006, Yoshida et al. 2010, Imoto et al. 2011) and the nucleus (Kato et al. 2011). In the mitotic phase, the cell shape drastically alters because organelle division occurs at the same time. Because of the change of cell shape, the mitotic phase of C. merolae cells can be easily distinguished using phase contrast microscopic observation. Because C. merolae is isolated from a hot spring, it adapts to a very harsh environment: pH 1.5 and 42°C. Hence, it is rare for other microorganisms to be contaminated during...
the cultivation of *C. merolae*. A recent study showed that *C. merolae* could harvest lipids (Kuroiwa et al. 2012). Expressing acyl-ACP reductase, *C. merolae* increases the production of lipids while maintaining cell proliferation (Sumiya et al. 2015). These results demonstrate that *C. merolae* has the potential to be an important model organism for studying algal biomass production, and that *C. merolae* will become important for industrial use.

In the process of using *C. merolae* as an industrial material, controlling its quality plays an important role. One of the most important aspects of controlling quality is measuring its division rate. Thus, we developed an algorithm to calculate this automatically. In this study, we developed an automatic classifier using a CNN algorithm for interphase and mitotic *C. merolae*.

### Materials and methods

#### Cultivation of *C. merolae*

*C. merolae* was maintained in 2×Allen’s media (Suzuki et al. 1994) under continuous white light at 42°C. For the synchronization of the cell cycle, *C. merolae* cells were subjected to a 2-h dark/12-h light/6-h dark cycle under aeration with ordinary air at 42°C. After synchronizing the culture, the *C. merolae* cells were collected and observed without fixation.

#### Image acquisition and processing

Phase contrast images were acquired under an upright microscope (BX53; Olympus, Japan) equipped with a DP72 digital camera (Olympus, Japan). The images were read using ImageJ software, and 64×64 pixel images of 50 interphase and 49 mitotic phase *C. merolae* were cropped from the read images. The images were saved in TIF format. The cropped images were read with the openCV functions as grayscale. Processing programs

---

### Table 1. Information on the layers of the models.

<table>
<thead>
<tr>
<th>Layer</th>
<th>1 conv.×1 pooling</th>
<th>2 conv.×2 pooling</th>
<th>3 conv.×3 pooling</th>
<th>4 conv.×4 pooling</th>
<th>Dropout function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution 1</td>
<td>1, 32, 3</td>
<td>1, 32, 3</td>
<td>1, 32, 3</td>
<td>1, 32, 3</td>
<td></td>
</tr>
<tr>
<td>Pooling 1</td>
<td>2×2</td>
<td>2×2</td>
<td>2×2</td>
<td>2×2</td>
<td></td>
</tr>
<tr>
<td>Convolution 2</td>
<td>—</td>
<td>32, 32, 3</td>
<td>32, 32, 3</td>
<td>32, 32, 3</td>
<td></td>
</tr>
<tr>
<td>Pooling 2</td>
<td>—</td>
<td>2×2</td>
<td>2×2</td>
<td>2×2</td>
<td></td>
</tr>
<tr>
<td>Convolution 3</td>
<td>—</td>
<td>—</td>
<td>32, 16, 3</td>
<td>32, 16, 3</td>
<td></td>
</tr>
<tr>
<td>Pooling 3</td>
<td>—</td>
<td>—</td>
<td>2×2</td>
<td>2×2</td>
<td></td>
</tr>
<tr>
<td>Convolution 4</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>16, 16, 3</td>
<td></td>
</tr>
<tr>
<td>Pooling 4</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>2×2</td>
<td></td>
</tr>
<tr>
<td>Linear 1</td>
<td>32768→5000</td>
<td>8192→500</td>
<td>1024→100</td>
<td>256→50</td>
<td></td>
</tr>
<tr>
<td>Linear 2</td>
<td>500→3</td>
<td>500→3</td>
<td>100→3</td>
<td>50→3</td>
<td></td>
</tr>
</tbody>
</table>

---

### Table 2. The result of the comparison of the number of the layers at the 100th training point.

<table>
<thead>
<tr>
<th>Model</th>
<th>Training accuracy</th>
<th>Test accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 conv.×1 pooling</td>
<td>1.00</td>
<td>0.82</td>
</tr>
<tr>
<td>2 conv.×2 pooling</td>
<td>1.00</td>
<td>0.89</td>
</tr>
<tr>
<td>3 conv.×3 pooling</td>
<td>1.00</td>
<td>0.90</td>
</tr>
<tr>
<td>4 conv.×4 pooling</td>
<td>1.00</td>
<td>0.84</td>
</tr>
</tbody>
</table>

---

### Table 3. The comparison of the number of the inserted dropout function in 2 conv. 2 pooling model.

<table>
<thead>
<tr>
<th>Number of dropout</th>
<th>Training accuracy</th>
<th>Test accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dropout 2</td>
<td>0.96</td>
<td>0.92</td>
</tr>
<tr>
<td>Dropout 3</td>
<td>1.00</td>
<td>0.89</td>
</tr>
</tbody>
</table>

---

### Table 4. The comparison of the number of the inserted dropout function in 3 conv. 3 pooling model.

<table>
<thead>
<tr>
<th>Number of dropout</th>
<th>Training accuracy</th>
<th>Test accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dropout 2</td>
<td>0.86</td>
<td>0.85</td>
</tr>
<tr>
<td>Dropout 3</td>
<td>0.82</td>
<td>0.81</td>
</tr>
<tr>
<td>Dropout 4</td>
<td>0.82</td>
<td>0.7</td>
</tr>
</tbody>
</table>

---

![Fig. 1. Scheme for the CNN model. After the images were input into the model, they were convolved and pooled in the layers repeatedly.](image-url)
were written in Python 3.7 bundled with Anaconda 3.0. The intensity of the images was normalized to 64. To obtain robust models, the images were increased by rotations of $0^\circ$, $90^\circ$, $180^\circ$, and $270^\circ$. Interphase images were tagged with zero and mitotic phase images were tagged with one. From each set of images, 75% were used as training models and 25% were used as test models.

**Construction of the training model**

The algorithms were written in Python 3.7 bundled with Anaconda 3.0. The programs were written on a Jupyter notebook 5.0.0 (http://jupyter.org/). To compose the CNN model, chainer (https://chainer.org/) was used. The images were processed using openCV functions. The convolution2d function was used to convolute the data, and max_pooling_2d and ReLU (rectified linear unit) functions in the chained functions were used to pool the data. The softmax-cross-entropy function was used for the loss function. Four patterns of models were composed, and the construction of the layers and convolution patterns are presented in Table 1. For all models, training was executed 100 times, and the accuracies of the training data and test data were recorded for all training executions. Accuracy was calculated as

$$\text{Accuracy} = \frac{\text{True counts}}{\text{True counts} + \text{False counts}}.$$

After we determined the number of layers, we compared the number of dropout functions.

**Results and discussion**

**Determination of the number of layers for the CNN module**

To construct a model that is highly accurate and robust, four patterns of models were tested to determine the number of layers (Fig. 1). Figure 2 shows the results of the training model evaluated using the training accuracy. The graphs indicate that the model classified the images accurately when the training and test accuracies were close to 1.0. By contrast, although the training accuracy achieved a higher score, the test accuracy achieved a lower score, and the model could not classify the unknown images substantially, which is referred to as overtraining or overfitting. Overtraining is a situation where the model can only classify the images used for training but cannot apply to new images. Figure 2a–d
show the results of the 1-, 2-, 3-, and 4-convolution and 4-pooling layer model, respectively. Table 2 shows the comparison of the training and test accuracy at the 100th training point. The result of 1-convolution and 1-pooling layer model and 4-convolution and 4-pooling layer model indicated that these models were lightly over-trained. Regarding these results, the 2-convolution and 2-pooling layer model and 3-convolution and 3-pooling layer model were suitable for classifying the interphase and mitotic phase *C. merolae*.

**Determination of the number of insertions of dropout functions**

To refine the model, we attempted to insert dropout functions into the 2-layer and 3-layer models and compared the accuracy of the number of inserted dropout functions. Using the dropout function, a part of nodes was randomly ignored during the process of training. Thus, the dropout function enabled to prevent the over biasing on minority nodes, resulting in avoidance of the overtraining (Srivastava et al. 2014). Figure 3 shows the results of the training model evaluated using the training accuracy and test accuracy, and Tables 3 and 4 show the comparison of the number of inserted dropout functions of 2-layer and 3-layer models at the 100th training point. Figure 3a and b show the 2-layer models, which inserted two and three dropout functions. Figure 3c–e show the 3-layer models, into which we inserted two, three and
four dropout functions, respectively. Regarding these results, we achieved the high accuracy of 92% with the 2-layer 2-dropout function model.

Acknowledgements

This research was supported by MXT/JSPS KAKENHI (15H05955 and 15H05962) for SM. We thank Maxine Garcia, Ph.D., from Edanz Group (www.edanzediting.com/ac) for editing a draft of this manuscript.

References


