Predicting the future direction of cell movement with convolutional neural networks

Shori Nishimoto[1], Yuta Tokuoka[1], Takahiro G Yamada[1],
Noriko F Hiroi[1, 2], and Akira Funahashi[1, *]

[1]Department of Biosciences and Informatics, Keio University,
Yokohama-shi, Kanagawa, 223-8522, Japan
[2]Faculty of Pharmaceutical Sciences, Sanyo-onoda City University,
Sanyo-Onoda, Yamaguchi, 756-0884, Japan
[*]Correspondence: funa@bio.keio.ac.jp

Summary

Image-based deep learning systems, such as convolutional neural networks (CNNs), have recently been applied to cell classification, producing impressive results; however, application of CNNs has been confined to classification of the current cell state from the image. Here, we focused on cell movement where current and/or past cell shape can influence the future cell fate. We demonstrate that CNNs prospectively predicted the future direction of cell movement with high accuracy from a single image patch of a cell at a certain time. Furthermore, by visualizing the image features that were learned by the CNNs, we could identify morphological features, e.g., the protrusions and trailing edge that have been experimentally reported to determine the direction of cell movement. Our results indicate that CNNs have the potential to predict the future cell fate from current cell shape, and can be used to automatically identify those morphological features that influence future cell fate.

Keywords

convolutional neural networks, deep learning, cell migration, cell fate prediction, image feature analysis, morphological feature

Introduction

Recent advances in microscope automation have enabled the acquisition of large numbers of bioimages. Several approaches to analyzing these images are proposed. One successful approach is machine learning, which has been used primarily in cell classification[1]. Cell classification using conventional machine learning proceeds in two steps. Firstly, hand-crafted image features are extracted for each cell from the image (e.g., using Scale-Invariant Feature Transform[2], Histograms of Oriented Gradients[3], or CellProfiler[4]). Secondly, these features are used to train a classification model (e.g., Support Vector Machine[5], Adaptive Boosting[6]). As a result, the performance of these classifiers relies heavily on the appropriateness of the hand-crafted features chosen empirically. Moreover, because feature extraction and classifier training are independent of each other, they cannot work together to identify and use discriminative information maximally.

In recent years, deep learning methods have been used to overcome this limitation of conventional machine learning methods. Deep learning methods, especially convolutional neural networks (CNNs), automatically learn feature representations from the raw pixels of cell images. Therefore, CNNs can avoid using hand-crafted features. Furthermore, CNNs are jointly optimized with these feature representations to predict the class for each cell image. For general visual recognition tasks, CNNs have substantially outperformed conventional machine learning methods with hand-crafted features[7, 8] , and they have been applied successfully to biological imaging[9, 10]. In cell classification, use of CNNs has produced impressive results[11, 12, 13, 14, 15, 16, 17] : e.g., the classification of abnormal morphology in MFC-7 breast cancer cells[13], the classification of cervical cells in cytology images[15]; the identification of malaria-infected cells[16]; and the automatic classification of Hep-2 (human epithelial-2) cell staining patterns[17].

The above applications of CNNs have focused on classification of the current cell state from the image. However, recent studies have demonstrated that the current and/or past cell shape influences the future cell fate even after the original shape is lost. Akamura et al. (2016) demonstrated that the shape of a V2 neural progenitor cell affects the stochastic fate decision-making of the daughter cells even after the V2 cell loses the original geometry through mitotic rounding and division[18]. Kozawa et al. (2016) used Bayesian inference to predict cell division-timing based on progressive cell-shape changes[19]. Therefore, an interesting question is whether CNNs can be used to predict future cell fate based on current and/or past cell shape.

Here, we focused on dynamic cell movement as a model system of cell shape influencing future cell fate. In general, cell movement can be conceptualized as a cyclic process[20]. The cell movement cycle begins with the formation of protrusions by actin polymerization. Protrusions are stabilized by adhering to the extracellular matrix (ECM). These adhesions serve as traction sites for the movement as the cell moves forward over them. At the cell rear, the adhesions with ECM are disassembled. Then the cell rear, the trailing edge, contracts mainly due to the pulling force generated by myosin II. As a result, the cell moves towards the protrusions. The motility and shape of individual migrating cells are closely related[21]. Jiang et al. (2005) demonstrated that the polarity of cell shape, i.e., wide front and a narrow rear, biases the direction of cell movement (i.e., moving direction)[22]. Ghosh et al. (2004) demonstrated that protrusions formed locally by actin polymerization define the moving direction[23]. We therefore hypothesized that CNNs can learn which morphological features influence moving direction and thus predict the future moving direction from cell shape at a certain time.

In general, it is not clear how and why CNNs arrive at a particular prediction decision, although several groups have attempted to interpret CNN predictions and propose possible methods for explaining CNNs decisions[24, 25, 26, 27]. These methods can visualize the features of the image that contribute to the CNN predictions. Application of these methods is expected to identify morphological features that influence moving direction.

Here, we demonstrate that CNNs prospectively predict the future direction of cell movement with high accuracy from a single image patch of a cell at a certain time. Furthermore, to reveal how and why CNN models can predict future moving direction, we visualized the features of the cell images that were learned by the CNN models and contributed to their prediction: e.g., the protrusions and trailing edge.

Results

Training and validation of CNNs for predicting the future direction of cell movement

To illustrate that CNNs can predict cell fate based on current cell shape, we set out to construct a CNN model that predicts the future direction of cell movement from a single image patch of a cell at a certain time. Firstly, we prepared image datasets from time-lapse phase-contrast microscopic images of migrating NIH/3T3 cells and U373 cells, respectively (see Materials and Methods). We manually tracked the positions of the migrating cells. Then, each cell in each frame was annotated with the moving direction: i.e., toward upper right, upper left, lower left, or lower right. The annotation was determined based on the displacement at the time the net displacement exceeded the average diameter of NIH/3T3 cells (18 μm*). For each annotated cell in each frame, we created an image patch of 128 × 128 pixels centered on its position coordinate. The NIH/3T3 dataset comprised 785 image patches for 40 cells (Table 1); the U373 dataset comprised 795 image patches for 12 cells (Table 1).

We used these datasets to train and test CNN models for predicting the direction of cell movement. In CNN models, input image patches are processed through multiple convolutional layers and max-pooling layers. In convolutional layers, trainable sets of filters are applied at different spatial locations using a stride of a certain size, thereby extracting features associated with moving direction as feature maps. In max-pooling layers, feature maps are down-sampled by computing the maximum value of a feature map over a region, which reduces

*http://bionumbers.hms.harvard.edu/bionumber.aspx?id=108905 variance and increases translational invariance[28]. After repeating the processing through convolutional layers and max-pooling layers, two fully connected layers are used for prediction. Our CNN models arrange 14 layers into eight convolutional layers, four max-pooling layers, and two fully connected layers, consisting of 162,704 trainable parameters in total (more detail in Material and Methods, network architecture shown in Fig. 1 and Table 2).

To assess the generalization performance of CNN models, we trained and tested them in 4-fold cross-validation using the NIH/3T3 and U373 datasets. We evaluated the performance of trained CNN models by calculating the average classification accuracy (ACA) and the mean class accuracy (MCA) (see Materials and Methods); the results are presented below as means ± standard deviation (n = 4 folds). While the expected value of ACA and MCA, if the data were randomly predicted, are 25%, our CNN models achieved high ACA of 87.3% ± 3.0% and high MCA of 85.9% ± 3.1% for the NIH/3T3 dataset. Our CNN models also achieved high ACA of 89.1% ± 0.4% and high MCA of 86.7% ± 1.1% for the U373 dataset.

Visualizing image features learned by the CNN models

Next, to identify the morphological features influential in the prediction of CNN models, we quantified and visualized the image features learned by the CNN models. Firstly, we used guided backpropagation (GBP)[27], which visualizes the local feature of the input image that most strongly activates a CNN's particular neuron (see Material and Methods). For each dataset, we presented each test image to the trained CNN models. Then, GBP was applied to a single maximum activation in each feature map of the last convolutional layer, producing local features of the input image patch. As a result, local features corresponding to the protrusion and the trailing edge contracting in the moving direction were identified for both the NIH/3T3 and U373 datasets (lower parts of each image group, Fig. 2). Next, we quantified and visualized global features of the image patches contributing to the CNN prediction of the moving direction by using deep Taylor decomposition (DTD)[26]. DTD quantifies the degree of contribution to the CNN prediction (relevance) of the input image patch (see Materials and Methods) on a pixel-by-pixel basis. For each dataset, we presented each test image to the trained CNN models. Then, by applying DTD to the prediction result, pixel-wise relevances were calculated. We visualized these pixel-wise relevances as heatmaps (Fig. 2, lower right). For both the NIH/3T3 and U373 datasets, pixels corresponding to the protrusion and trailing edge simultaneously contributed to the prediction of the CNN models.

Discussion

Previous applications of CNNs to cell classification have focused on classification of the current cell state from an image[11, 12, 13, 14, 15, 16, 17]. In contrast, here, we focused on images of dynamic cell movement and demonstrated that

CNNs can prospectively predict the future direction of cell movement with high accuracy.

Average speed and directionality are important parameters that define cell motility[29]. Both these parameters were significantly different between the NIH/3T3 and the U373 datasets (Supplementary Fig. 1). The apparent morphology of migrating cells also differed qualitatively between the two datasets: e.g., the protrusions were prominent and broad in U373 cells compared with NIH/3T3 cells (Fig. 2, Movie 1 and 2 in Supplementary Material). Despite these differences, our CNN models achieved the same degree of prediction accuracy for both cell types. This might be because our CNN models could learn the morphological features, such as the protrusion and the trailing edge (Fig. 2), which are known to be characteristic features of cell movement regardless of cell type or substrate, except for some specific cell types[20, 30].

Because cell classification using conventional machine learning is built upon the extraction of hand-crafted features[1, 2, 3, 4], the features affecting the classification result depend on the choice of hand-crafted features. These hand-crafted features are abstract representations of the cell image and are therefore diffcult to intuitively relate to visual inspection of the cell image. In contrast, because CNNs automatically and jointly learn optimal feature representations and the prediction task directly from the raw pixels of the cell image, the features affecting the prediction result do not depend on the feature selection and are optimized for the prediction. Furthermore, because these features can be visualized on the cell image by using the approach used in this study, they can be intuitively related to visual inspection of the cell image. Here, we identified morphological features, such as protrusions and trailing edge, which are known experimentally to influence the moving direction[for review, see [20]]: for instance, protrusions generated locally by actin polymerization determine the moving direction[23], and the position of the trailing edge influences the moving direction via stress fiber organization[31]. Our results indicate that such important morphological features that can influence cell fate can be identified automatically, by using CNNs to predict cell fate and then visualizing the image feature(s) contributing to this prediction.

In conclusion, we focused on dynamic cell movement as a model system in which cell shape influences cell fate. Our findings indicate that use of CNNs can enable not only the prediction of the cell fate but also the identification of morphological features that can influence the cell fate. Based on these findings, we believe that our approach could be applied to the analysis of other cell mechanisms where cell shape can influence future processes, such as cell division[19] and differentiation[32]. This approach could reveal unknown biological phenomena and be useful for pathologic and/or clinical diagnosis.

Acknowledgements

We thank the Doi laboratory at Keio University for providing NIH/3T3 fibroblasts. We thank Takumi Hiraiwa from the Department of Biosciences and

Informatics at Keio University for assistance with experiments. We are also grateful to Carlos Ortiz-de-Solorzano from the Center for Applied Medical Research (Pamplona, Spain) for providing the dataset used in the ISBI cell tracking challenge 2015. The research was funded by a JSPS KAKENHI Grant (Number 16H04731).

Author contributions

A.F. initiated the project; A.F. and N.H. supervised the project; S.N. prepared time-lapse phase-contrast microscopy images of NIH/3T3 cells, created the datasets, developed the convolutional neural network approach, analyzed the data, and wrote the manuscript with Y.T. and T.Y.; A.F. and Y.T. gave technical advice on analysis; N.H. advised from a biological point of view. All authors commented on the manuscript.

Declaration of Interests

The authors declare no competing interests.

References

[1] Christoph Sommer and Daniel W Gerlich. Machine learning in cell biology–teaching computers to recognize phenotypes. *J Cell Sci*, 126(24):5529–5539, 2013.

[2] DavidG. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 1:886–893 vol. 1, June 2005.

[4] Anne E Carpenter, Thouis R Jones, Michael R Lamprecht, Colin Clarke, In Han Kang, Ola Friman, David A Guertin, Joo Han Chang, Robert A Lindquist, Jason Moffat, et al. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7(10):R100, 2006.

[5] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[6] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August 1997.

[7] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE, 2014.

[8] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[9] Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. Deep learning for computational biology. *Molecular systems biology*, 12(7):878, 2016.

[10] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. Deep learning in bioinformatics. *Briefings in Bioinformatics*, page bbw068, 2016.

[11] B. Jiang, X. Wang, J. Luo, X. Zhang, Y. Xiong, and H. Pang. Convolutional neural networks in automatic recognition of trans-differentiated neural progenitor cells under bright-field microscopy. In *2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)*, pages 122–126, Sept 2015.

[12] Oliver Dürr and Beate Sick. Single-cell phenotype classification using deep convolutional neural networks. *Journal of Biomolecular Screening*, 21(9):998–1003, 2016. PMID: 26950929.

[13] Oren Z Kraus, Jimmy Lei Ba, and Brendan J Frey. Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics*, 32(12):i52–i59, 2016.

[14] David A Van Valen, Takamasa Kudo, Keara M Lane, Derek N Macklin, Nicolas T Quach, Mialy M DeFelice, Inbal Maayan, Yu Tanouchi, Euan A Ashley, and Markus W Covert. Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. *PLoS Comput Biol*, 12(11):e1005177, 2016.

[15] Ling Zhang, Le Lu, Isabella Nogues, Ronald Summers, Shaoxiong Liu, and Jianhua Yao. Deeppap: Deep convolutional networks for cervical cell classification. *IEEE Journal of Biomedical and Health Informatics*, 2017.

[16] Yuhang Dong, Zhuocheng Jiang, Hongda Shen, W David Pan, Lance A Williams, Vishnu VB Reddy, William H Benjamin, and Allen W Bryan. Evaluations of deep convolutional neural networks for automatic identification of malaria infected cells. In *Biomedical & Health Informatics (BHI), 2017 IEEE EMBS International Conference on*, pages 101–104. IEEE, 2017.

[17] Zhimin Gao, Lei Wang, Luping Zhou, and Jianjia Zhang. Hep-2 cell image classification with deep convolutional neural networks. *IEEE journal of biomedical and health informatics*, 21(2):416–428, 2017.

[18] Takashi Akanuma, Cong Chen, Tetsuo Sato, Roeland MH Merks, and Thomas N Sato. Memory of cell shape biases stochastic fate decision-making despite mitotic rounding. *Nature communications*, 7, 2016.

[19] Satoshi Kozawa, Takashi Akanuma, Tetsuo Sato, Yasuomi D Sato, Kazushi Ikeda, and Thomas N Sato. Real-time prediction of cell division timing in developing zebrafish embryo. *Scientific reports*, 6, 2016.

[20] Anne J Ridley, Martin A Schwartz, Keith Burridge, Richard A Firtel, Mark H Ginsberg, Gary Borisy, J Thomas Parsons, and Alan Rick Horwitz. Cell migration: integrating signals from front to back. *Science*, 302(5651):1704–1709, 2003.

[21] Alex Mogilner and Kinneret Keren. The shape of motile cells. *Current Biology*, 19(17):R762–R771, 2009.

[22] Xingyu Jiang, Derek A Bruzewicz, Amy P Wong, Matthiẽu Piel, and George M Whitesides. Directing cell migration with asymmetric micropatterns. *Proceedings of the National Academy of Sciences*, 102(4):975–978, 2005.

[23] Mousumi Ghosh, Xiaoyan Song, Ghassan Mouneimne, Mãzen Sidani, David S Lawrence, and John S Condeelis. Cofilin promotes actin polymerization and defines the direction of cel l motility. *Science*, 304(5671):743–746, 2004.

[24] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[25] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[26] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.

[27] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[28] Fu Jie Huang, Y-Lan Boureau, Yann LeCun, et al. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[29] Atsuko Kodama, Iakowos Karakesisoglou, Ellen Wong, Alec Vaezi, and Elaine Fuchs. Acf7: an essential integrator of microtubule dynamics. *Cell*, 115(3):343–354, 2003.

[30] Miguel Vicente-Manzanares, Donna J Webb, and A Rick Horwitz. Cell migration at a glance. *Journal of cell science*, 118(21):4917–4919, 2005.

[31] Raphaela Rid, Natalia Schiefermeier, Ilya Grigoriev, J Victor Small, and Irina Kaverina. The last but not the least: the origin and significance of trailing adhesions in fibroblastic cells. *Cytoskeleton*, 61(3):161–171, 2005.

[32] Rowena McBeath, Dana M Pirone, Celeste M Nelson, Kiran Bhadriraju, and Christopher S Chen. Cell shape, cytoskeletal tension, and rhoa regulate stem cell lineage commitment. *Developmental cell*, 6(4):483–495, 2004.

[33] Martin Maška, Vladimír Ulman, David Svoboda, Pavel Matula, Petr Matula, Cristina Ederra, Ainhoa Urbiola, Tomás España, Subramanian Venkatesan, Deepak MW Balak, et al. A benchmark for comparison of cell tracking algorithms. *Bioinformatics*, 30(11):1609–1617, 2014.

[34] Vladimír Ulman, Martin Maška, Klas EG Magnusson, Olaf Ronneberger, Carsten Haubold, Nathalie Harder, Pavel Matula, Petr Matula, David Svoboda, Miroslav Radojevic, et al. An objective comparison of cell-tracking algorithms. *Nature methods*, 14(12):1141, 2017.

[35] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[37] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[39] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[40] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*, volume 5, 2015.

Figures

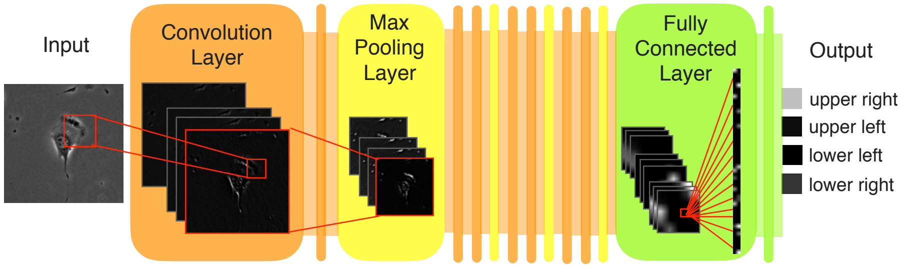

Figure 1: **Architecture of CNN models that predict the future direction of cell movement using a single image patch of a cell at a certain time.** In the flowchart, an image patch of a NIH/3T3 cell annotated on the upper right (Input) is presented to a CNN model. The input is processed through a series of repeating convolutional layers (orange) and max-pooling layers (yellow). In the convolutional layer, the activation images illustrate extracted feature maps of the sample image patch (Input). The red boxes and lines illustrate the connections within the CNN model. After repeating processing through convolutional layers and max-pooling layers, fully connected layers are used for prediction (green). The network output (Output) represents the distribution over four moving directions.

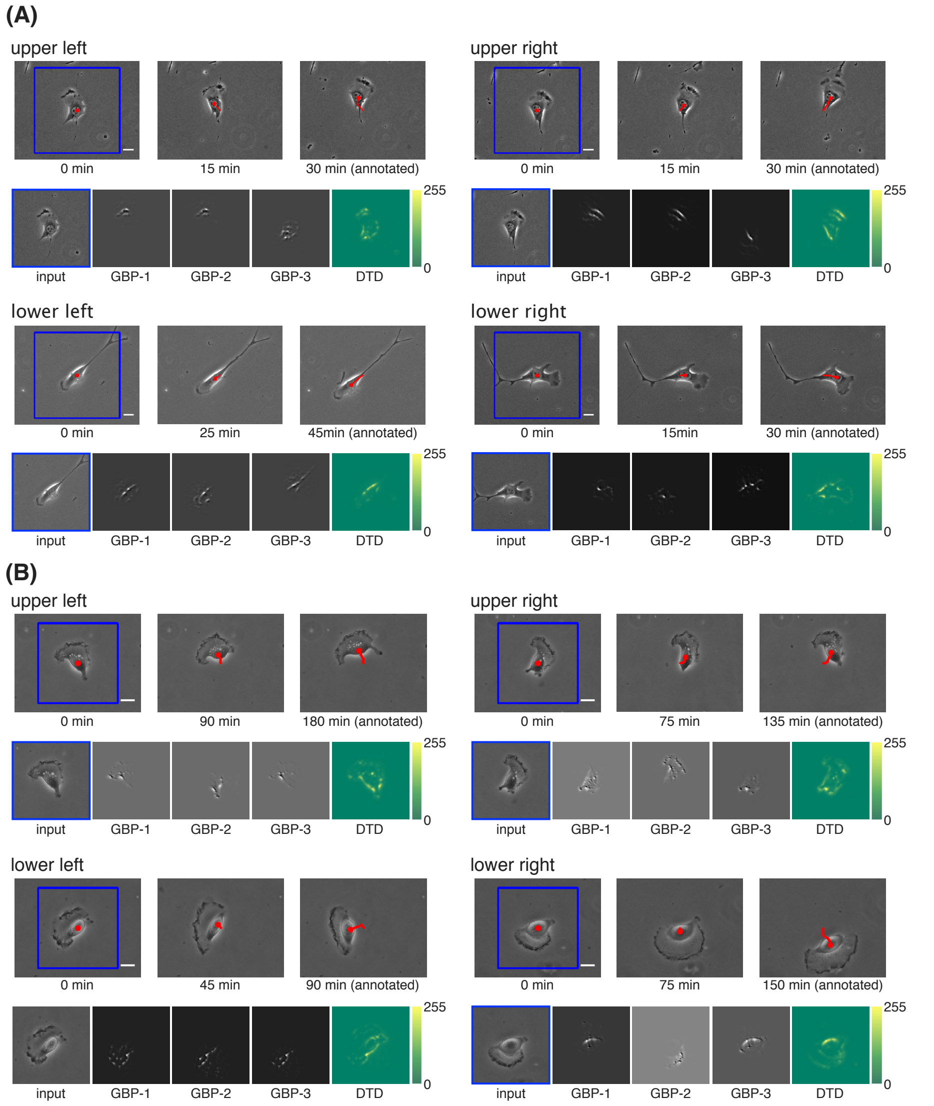

Figure 2: **Visualized image features learned by the CNN models.** (A) NIH/3T3 dataset. (B) U373 dataset. For each moving direction, each group of images shows exemplary results (i.e., those for a correctly predicted test image patch). The upper row of each group of images comprises—from left to right—the frame corresponding to the input image patch, the frame imaged in the middle between the left frame and the right frame, and the frame when the moving direction was annotated (scale bars, 20 μm). The time under each frame shows the elapsed time since the leftmost frame was imaged. The blue bounding box indicates the area corresponding to the input image patch. The red dot indicates the position of the cell obtained by manual tracking. The red line indicates the trajectory of cell movement starting from the position of the cell at 0 min. The lower row of each group of images comprises—from left to right—the input image patch, the local features visualized by GBP for the feature maps whose maximum activations were top three, and the heatmap of pixel-wise relevance calculated by DTD.

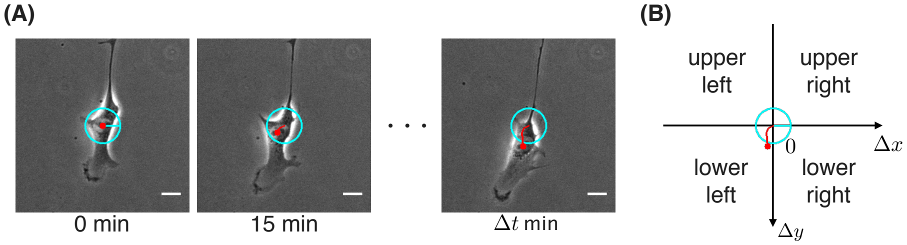

Figure 3: **Annotation of the moving direction.** (A) Exemplary time-lapse images of a migrating cell (scale bars, 20 μm). The time under each frame shows the elapsed time since the leftmost frame (annotation target) was imaged. Δt is the time when the net displacement $\sqrt{\Delta x^2 + \Delta y^2}$ exceeded the average diameter of NIH/3T3 cells. The red dot indicates the position of the cell obtained by manual tracking. The red line indicates the trajectory of cell movement starting from the position of the cell at 0 min. The radius of the cyan circle is the average diameter of NIH/3T3 cells. (B) Annotating one of the four moving directions. According to the value of cell displacement $(\Delta x, \Delta y)$ at the time Δt, the moving direction was annotated as shown in the figure. The red line and the cyan circle are the same as those of the frame at Δt min in (A).

Tables

Table 1: Number of image patches per moving direction

Moving direction	Number of image patches in the NIH/3T3 dataset	Number of image patches in the U373 dataset
upper right	179	165
upper left	188	253
lower left	242	96
lower right	176	281

Table 2: CNN model-relevant hyperparameters

Layer	Type	Description
1	Convolution	Filter size = 3 × 3, Number of filters = 16, Stride size = 1
2	Convolution	Filter size = 3 × 3, Number of filters = 16, Stride size = 1
3	Max-pooling	Filter size = 2 × 2, Stride size = 2
4	Convolution	Filter size = 3 × 3, Number of filters = 16, Stride size = 1
5	Convolution	Filter size = 3 × 3, Number of filters = 16, Stride size = 1
6	Max-pooling	Filter size = 2 × 2, Stride size = 2
7	Convolution	Filter size = 3 × 3, Number of filters = 32, Stride size = 1
8	Convolution	Filter size = 3 × 3, Number of filters = 32, Stride size = 1
9	Max-pooling	Filter size = 2 × 2, Stride size = 2
10	Convolution	Filter size = 3 × 3, Number of filters = 64, Stride size = 1
11	Convolution	Filter size = 3 × 3, Number of filters = 64, Stride size = 1
12	Max-pooling	Filter size = 2 × 2, Stride size = 2
13	Fully connected	Number of neurons = 100
14	Fully connected	Number of neurons = 4

Materials and Methods

Time-lapse phase-contrast microscopic images

To create datasets for training CNN models, we prepared time-lapse phase-contrast microscopic images of cell movement in different ways for each dataset; details are provided below.

NIH/3T3 dataset

NIH/3T3 fibroblasts were plated on glass-based dishes precoated with 5 μg/cm^2 fibronectin at a density of 500 cells/cm^2. After overnight incubation, cell movements were monitored with an inverted microscope (IX81, Olympus) equipped with an on-stage incubation chamber that maintained the temperature at 37°C and the CO_2 concentration at 5%, using a 20× objective (0.45 numerical aperture). Phase-contrast images were collected with CCD video cameras (ORCA-Flash4.0 or ORCA-R2; Hamamatsu) at 5-min intervals, digitized and stored as image stacks by using Micro-Manager software (Open Imaging).

U373 dataset

Time-lapse phase-contrast microscopic images of glioblastoma–astrocytoma U373 cells were obtained from the dataset used in the ISBI (International Symposium on Biomedical Imaging) cell tracking challenge 2015[33, 34]. U373 cells moving on a polyacrylamide substrate were monitored with a microscope (Nikon) using a 20× objective (0.5 numerical aperture), and phase-contrast images were acquired at 15-min intervals.

Annotation of the moving direction

For each dataset, we manually tracked the positions of migrating cells by using the Manual Tracking plugin of ImageJ (National Institutes of Health). For each cell in each frame, we annotated one of the four moving directions according to the value of displacement $(\Delta x, \Delta y)$ at the time the net displacement $\sqrt{\Delta x^2 + \Delta y^2}$ (the distance between the initial and final positions) exceeded the average diameter of NIH/3T3 cells (18 μm[†])(Fig. 3). The four moving directions were defined as follows: (i) $\Delta x \geq 0, \Delta y \leq 0 =$ upper right; (ii) $\Delta x < 0, \Delta y \leq 0$ = upper left; (iii) $\Delta x < 0, \Delta y > 0 =$ lower left; and (iv) $\Delta x \geq 0, \Delta y > 0 =$ lower right. Regarding the NIH/3T3 dataset, the net displacement was evaluated at 15-min intervals according to the shooting interval of the U373 dataset. For each annotated cell, we cropped the image to a bounding box of 256 × 256 pixels for the NIH/3T3 dataset or 170 × 170 pixels for the U373 dataset, centered on the cell x, y coordinates. Cropped image patches were resized to 128 × 128 pixels for each dataset.

[†] http://bionumbers.hms.harvard.edu/bionumber.aspx?id=108905

CNNs for predicting the future direction of cell movement

To prospectively predict the future direction of cell movement from a single image patch of a cell at a certain time, we used CNN models[35, 36] with the architecture shown in Fig. 1 and Table 2. Each of our CNN models comprised consecutive convolutional layers, max-pooling layers, and fully connected layers. The last fully connected layer consisted of four neurons, with each neuron corresponding to the future direction of cell movement. We applied the softmax function to the activations of the last fully connected layer to produce a distribution over four moving directions. Each convolutional layer and the first fully connected layer was followed by a nonlinear activation function. We used rectified linear units, which have been shown to introduce nonlinearities without suffering from the vanishing gradient problem[37]. To enable the CNN models to predict the future moving direction for image patches with various intensity values, the pixel values of each input image patch were normalized to within the range of 0 to 1 by first subtracting the minimum intensity value of the image patch, and then dividing by the maximum intensity value of the resulting image patch[17].

Training and validation of CNN models

We trained CNN models and tested them in 4-fold cross-validation. At each fold, we used a softmax loss function and trained the CNN model for 50 epochs, using stochastic gradient descending with stratified batches of 32 images. To reduce the influence of imbalanced moving directions in the datasets, we multiplied the loss function by the class weight expressed by the following equation:

$$\frac{\sum_k^n N_k}{nN_k}$$

where n is the number of classes (moving directions), and N_k is the number of training data of class k. We initialized all weights in a CNN model by using the HeNormal algorithm, which sets the weights of each layer according to a scaled Gaussian distribution whose standard deviation corresponds to the input size of each layer[38]. We used standard values for the base learning rate (0.01) and momentum (0.9). For each epoch, we evaluated the prediction accuracy of a CNN model using test data. The evaluation criteria were the ACA and the MCA: ACA is the overall correct prediction rate of all test data, and MCA is the average of the per-class accuracies defined as follows:

$$\mathrm{MCA} = \frac{1}{n}\sum_{k=1}^{n} \mathrm{CCR}_k$$

where CCR_k is the prediction accuracy of class (moving direction) k and n is the number of classes. If the MCA was not improved for the best score in the previous epoch, the learning rate was decreased by multiplying by 0.9. To prevent the CNN models from overfitting to training data, we (a) trained the models using vertical and horizontal reflections of the image patches as well as rotations by $90°, 180°$, and $270°$, in addition to the original training data; and (b) applied dropout[39] to the first fully connected layer (dropout rate, 0.5). For each fold, we used the best score of each criterion for 50 epochs as the prediction accuracy of the models.

For the following analysis of image features learned by CNN models, we used those models with the best MCA for the test data at each fold. We implemented and trained the CNN models by using Chainer[40], which is open-source software for machine learning.

Visualizing image features learned by CNN models

We quantified and visualized features learned by the CNN models as described in the Results. GBP[27] was used to visualize local features of input image patches specific to the annotated moving direction. In this method, an input image is presented to a CNN, and high-level feature maps are computed throughout the layers. Typically, a single neuron activation is left non-zero in the high-level feature map. The single non-zero activation is back-propagated to input pixel space. Finally, the part of the input image that is most strongly activating the single non-zero neuron (i.e., the part that is most discriminative) is reconstructed. For each dataset, we presented each test image patch to trained CNN models, and a single maximum activation in each feature map of the last convolutional layer was back-propagated to input pixel-space, resulting in producing local features of the input image.

To quantify and visualize global features of input image patches contributing to the CNN prediction of moving direction, we used DTD[26], which can be summarized as follows. As a premise, each neuron of a CNN is viewed as a function that can be expanded and decomposed on its input variables. A CNN prediction of the moving direction is obtained by forward-propagation of input pixel values $\{x_p\}$ and is encoded by the output neuron x_f. The output neuron is assigned a relevance score $R_f = x_f$ representing the total evidence for the specific moving direction. Relevance is then decomposed and back-propagated from the top layer down to the input. As a result, the pixel-wise relevance (i.e., the degree of contribution to the CNN prediction) is calculated for each pixel of the input image. The pixel-wise relevances can be visualized as a heatmap. For each dataset, we presented each test image patch to trained CNN models, and DTD was applied to the CNN prediction output.

Data availability

The image datasets used to train the CNN models, the raw time-lapse phase contrast images, metadata text files of Micro-Manager, and tracking results are available using bash scripts in: https://github.com/funalab/PredictMovingDirection.

Code availaility

The code for performing the experiments is available at: https://github.com/funalab/PredictMovingDirection.

Supplementary Material

Supplementary Figure 1. Motility values calculated for the NIH/3T3 and U373 datasets. (A) Boxplot of the average speed of the cell, v. (B) Boxplot of the directionality of the cell, k ($n = 785$ cell images in the NIH/3T3 dataset and 795 cell images in the U373 dataset). P-value is from two-sided Mann-Whitney rank test. For each image in the datasets, we first measured the time required until the net displacement Δr exceeded the average diameter of NIH/3T3 cells, which is the time interval Δt until the moving direction was annotated. Then, we calculated the total distance $\sum \Delta d$ traveled by the cell in the time interval Δt. Regarding the NIH/3T3 dataset, the net displacement Δr and movement distance Δd were calculated at 15-min intervals according to the shooting interval of the U373 dataset. The average speed was calculated by the equation $v = \sum \Delta d / \Delta t$[22]. The directionality was calculated by dividing the net displacement Δr by the total distance $\sum \Delta d$[22]. The directionality was used to measure how often the cell tended to turn. Cells that frequently make turns will yield a k value close to 0, whereas cells that persistently move along one direction will yield a k value close to 1.

Movie 1. Time-lapse phase-contrast images of migrating NIH/3T3 cells used to create the NIH/3T3 dataset (AVI). Images were acquired as described in Materials and Methods.

Movie 2. Time-lapse phase-contrast images of migrating U373 cells used to create the U373 dataset (AVI). Images were acquired as described in Materials and Methods.