Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 54-20 (2021) 328–333

Actin Cytoskeleton Morphology Modeling Using Graph Embedding and Classification in Machine Learning ⋆

Yi Liu * **Juntao Zhang** * **Charuku Bharat** * **Juan Ren** *

* *Iowa State University, Ames, IA 50011 USA (e-mail: juanren@iastate.edu)*

Abstract: Actin cytoskeleton modeling and quantification are essential in studying the dynamics of cellular mechanotransduction. However, current approaches to actin cytoskeleton quantification are limited in terms of both efficiency and accuracy. In this paper, we propose to model the cellular actin cytoskeleton morphology using the graph to vector embedding technique together with the neural network (NN) classification in machine learning. The proposed model consists of a skip-gram model followed by a fully connected classifier. The actin cytoskeleton morphology is modeled based on both the structure and node features extracted from the cytoskeleton images. Specifically, the embedding tool outputs the embedded vectors of the cytoskeleton graphs, and then the embedded vectors are used by the fully connected layer to perform cytoskeleton classification. In this work, we demonstrate the classification accuracy of the proposed framework using actin cytoskeleton images from cells treated by Latrunculin B (an actin depolymerizer) at different concentrations. The actin cytoskeleton morphology corresponding to each treatment concentration is defined as a class (e.g., actin depolymerization level). The final classification result is showed an accuracy of 85.3%.

Keywords: Graph to vector embedding, Skip-gram model, Classification, Machine learning, Actin cytoskeleton.

1. INTRODUCTION

Accumulating evidence indicates that the cellular cytoskeleton has significant abilities to affect and reflect the numerous physiological state and properties of cellular mechanotransduction (Lichtenstein et al., 2003; Mollaeian et al., 2018). As one of the main kinds of cytoskeletal filaments, actin cytoskeleton (or actin filaments) are able to control the cellular mechanotransduction properties, such as elasticity, poroelasticity, adhesion, cell morphology, etc., through affecting motility, contractility, and dynamic stability of cells (Lichtenstein et al., 2003; Mollaeian et al., 2018). Particularly, the actin cytoskeleton reshapes in order to regulate cellular dynamic changes triggered by external stimuli, such as the externally applied force and the change of substrate stiffness (Gupta et al., 2015; Wakatsuki et al., 2001; Sims et al., 1992). Therefore, the actin cytoskeleton morphology encodes essential information of the cellular dynamic properties. However, effectively recognizing the morphology of actin cytoskeleton and investigating its correlation with cellular biomechanical properties remains challenging.

Takumi et al. (2010) developed an image analysis framework to quantify the cytoskeleton orientation, bundling, and density using the measurement of fluorescence microscopic images of plant cells (Higaki et al., 2010). Kimori et al. (2016) used a mathematic method to quantify the morphological properties of biological structures of the actin cytoskeleton in the plant cells (Kimori et al., 2016). We also proposed an image recognition-based approach to quantify the actin cytoskeleton using edge detector, Hough transforms, and Matlab filling tools (Liu et al., 2018). However, these methods all require manual processing and analysis of cellular cytoskeleton images, which often feature extensive protrusions and blebs that are difficult to distinguish using traditional parameters. Recent advances in image modeling with neural networks have provided a way to derive representations of cell shapes and cellular structures.

Osokin et al. (2017) proposed a novel application of Generative Adversarial Networks (GAN) to the synthesis of cell images taken by fluorescence microscopy (Osokin et al., 2017). Alex et al. (2019) introduced a convolutional neural network (CNN) using the self-supervised method to learn feature representations of single cells in fluorescent microscopy images without labeled training data(Lu et al., 2019). Jude Phillip et al. (2021) presented an unsupervised machine learning method for the analysis of cell and nuclear morphology from bright-field or fluorescent images using the Visually Aided Morpho-Phenotyping Image Recognition (VAMPIRE) algorithm, which was designed by Pei-Hsun et al. (2015) (Wu et al., 2015) and enabled the profiling and classification of cells based on the cell boundary points and nuclear contours (Phillip et al., 2021).

However, these methods either only focused on external cell morphology (Osokin et al., 2017; Lu et al., 2019) or

---

⋆ This work was supported by the National Science Foundation (NSF) CMMI-1634592 and CMMI-1751503, and Iowa State University.

10.1016/j.ifacol.2021.11.195

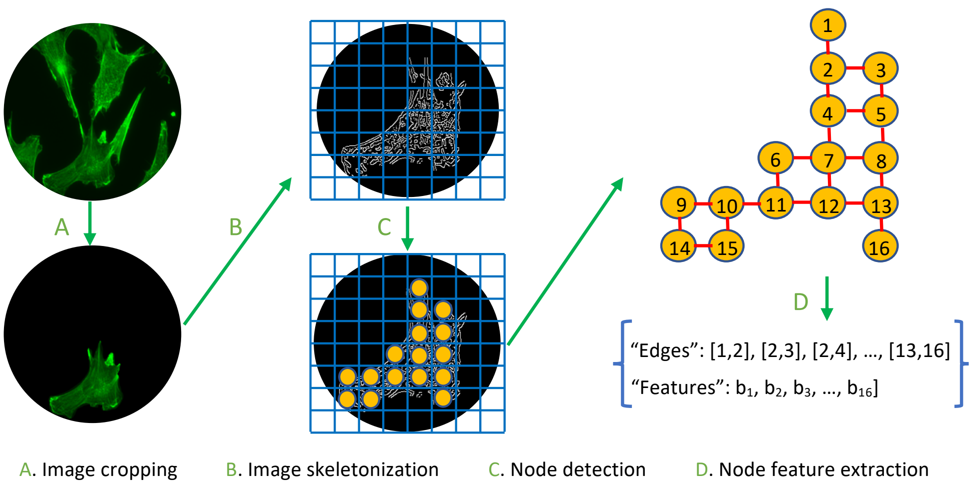

Fig. 1. Preprocessing process of the fluorescent actin cytoskeleton image. Single cell images were cropped from the original ones. Graph nodes are generated and labeled based on the pixel brightness. The list consisting edges between neighbor nodes on four directions and features which are the node brightness represents the actin cytoskeleton graph.

required a high-throughput cell imaging platform (htCIP) that provides access to extracting high-content individual cells with cellular and nuclear morphology (Phillip et al., 2021). Therefore, this study aims to develop a new framework of deep learning to extract feature representations of the cellular actin cytoskeleton with high-throughput and accuracy.

In this paper, we propose to model the cellular actin cytoskeleton morphology using the graph to vector embedding technique together with the neural network classification in machine learning. The proposed model consists of a skip-gram model followed by a fully connected classifier. The actin cytoskeleton morphology is modeled based on both the structure and node features extracted from the cytoskeleton images. Specifically, the embedding tool outputs the embedded vectors of the cytoskeleton graphs, and then the embedded vectors are used by the fully connected layer to perform cytoskeleton classification. In this paper, we demonstrate the classification accuracy of the proposed framework using actin cytoskeleton images from cells treated by Latrunculin B at different concentrations. The actin cytoskeleton morphology corresponding to each treatment concentration is defined as a class (e.g., actin depolymerization level). The final classification result is showed an accuracy of 85.3%.

2. DEEP LEARNING FRAMEWORK FOR ACTIN CYTOSKELETON MODELING

2.1 Convert Images to Graphs

Single-cell fluorescence images are used in the proposed framework for actin cytoskeleton classification. The color of the pixels outside the detected cell area can be mandatorily set as black to remove the background color. To process the single-cell actin cytoskeleton images, the original RGB images need to be converted to grayscale in which the range of the brightness for each pixel was from 0∼255 (Ojala et al., 2002), the pixels lower than the average brightness were set as zero to minimize the noise effect (Wang et al., 1999). To convert the actin cytoskeleton images to actin cytoskeleton graphs, a Canny edge detector is used to complete the actin fiber skeletonization (Bao et al., 2005). For the skeletonized images, each non-zero pixel (pixel with brightness larger than the average brightness) is treated as a node with its brightness as its node feature. Graph nodes are explored from one randomly selected root node through four directions (i.e., up, down, left, and right) using Breadth-first search. Edges created between two adjacent nodes and node brightness are recorded to represent the structure and feature of actin cytoskeleton graph G as shown in Fig. 1.

2.2 Sampling and Relabeling all Subgraphs from the graph

Subgraph is a set of nodes that appears around the selected root node n. Nodes in a subgraph are not further than the designed distance (or walk depth) from the root node, see Fig. 2. To sample and label subgraphs, we follow the well-known Weisfeiler-Lehman (WL) relabeling process (Shervashidze et al., 2011), which lays the basis for the WL kernel (Yanardag and Vishwanathan, 2015; Shervashidze et al., 2011). The subgraph extraction process is shown in

Algorithm 1 GetSubgraph

```
input  : $n$: The root node of the subgraph
         $G = (N, E, \lambda)$: Given graph
         $d$: Walk depth considered in subgraph extraction

output: $sg_n^{(d)}$: Rooted subgraph of walk depth $d$ around
         node $n$
Function GetSubgraph($n, G, d$):
    $sg_n^{(d)} = \{\}$ if $d = 0$ then
        $sg_n^{(d)} := \lambda(n)$
    end
    else
        $N_n := \{n' \mid (n, n') \in E$  $M_n^d := \{$GetSubgraph
         $(n', G, d-1) \mid n' \in N_n\}$  $sg_n^{(d)} := sg_n^{(d)} \cup$ GetSub-
         graph $(n, G, d-1) \oplus sort(M_n^d)$
    end
    return $sg_n^{(d)}$
```

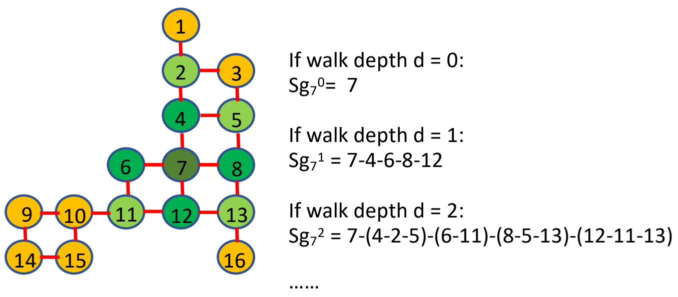

Fig. 2. Subgraphs sg_n of node $n = 7$ with walk depth d in range from $0 \sim 2$.

Algorithm 1. The algorithm takes a given graph G from which the subgraph needs to be extracted, a root node n, and the walk depth d of the intended subgraph as inputs and returns the extracted subgraph $sg_n^{(d)}$.

A walk depth d is applied around the root node n in a given graph G. If $d = 0$, no subgraph exists, therefore the label of node n is returned. For cases when $d > 0$, all the neighbors n' of n need to be explored. For each neighbor node, walk depth $d - 1$ is applied to extract subgraphs of n'. All subgraphs are saved in list M_n^d. At the end, the walk depth $d - 1$ subgraph around the root node n is obtained and then concatenated with sorted list M_n^d to achieve the subgraph $sg_n^{(d)}$.

2.3 Graph embedding

In order to reduce the dimension of input data, fluorescent images of the actin cytoskeleton are embedded in vectors based on their network structure and fluorescent brightness, which correspond to the actin cytoskeleton alignment and quantity, respectively. The idea of embedding is to train the weight matrix of the hidden layer to find efficient representation for given graphs (Jurafsky and Martin, 2000), see Fig. 3.

Algorithm 2 Graph embedding

```
input  : $\mathbb{G} = \{G_1, G_2, ..., G_n\}$: Set of graphs
         $D$: Maximum walk depth of rooted subgraphs
         $\delta$: Embedding size
         $\mathbf{e}$: Number of epochs
         $\alpha$: Learning rate

output: Embedded vector of graphs $\Phi \in R^{|\mathbb{G}|\times\delta}$
Function Graph2Vec($\mathbb{G}, D, \delta, \mathbf{e}, \alpha$):
    Initialization: Initialize weight matrix $\Phi \in R^{|\mathbb{G}|\times\delta}$ for $e = 1$ to e do
        $\mathcal{G} = SHUFFLE(\mathbb{G})$ for $G_i = (N_i, E_i, \lambda_i) \in \mathcal{G}$ do
            for $n \in N_i$ do
                for $d$ = 0 to $D$ do
                    $sg^{(d)}$ := GetSubgraph $(n, G_i, d)$
                    $J(\Phi) = -logPr(sg^{(d)}|\Phi(G_i))$
                    $\Phi = \Phi - \alpha\frac{\partial J}{\partial \Phi}$
                end
            end
        end
    end
    return $\Phi$
```

In order to embed the actin cytoskeleton graphs, the skip-gram model is applied as the embedding layer, which is commonly used in Word2Vec embedding and yields the highest overall accuracy, and consistently produces the highest accuracy on semantic relationships (Mikolov et al., 2013). The skip-gram model is trained to maximize the probability of predicting subgraphs that exist in the graph that needs to be embedded on the input, see **Algorithm 2**. In the beginning, the weight matrix Φ (i.e., embedding vector) is initialized with random values. The given graph $G_i \in \mathbb{G}$ is fed as a one-hot vector (Harris and Harris, 2010) in $\mathbf{e}$ epochs to fit the subgraphs contained in graph G_i. During the embedding process, actin cytoskeleton images with similar structure and quantity are embedded closer than the others. Specifically, cell images taken under the same conditions can be represented using vectors having a closer distance in vector space, which improves the later classification accuracy. Since subgraphs set $SG_{vocab} = \{sg_1, sg_2, ...\}$ are extracted from each node of each graph, its size is very large. This means for each input, it only makes very small changes to the huge amount of weights even though there exists just one true example. This makes the training process very inefficient. Therefore, we propose to use negative sampling which approximates the loss from the softmax layer by updating a small subset of all the weights at once, i.e., update the weights of the correct label but only a small number of the incorrect label (Goldberg and Levy, 2014). This makes the network training more efficient.

2.4 Classification Model Architecture

After the embedded actin cytoskeleton graph vectors were obtained, they were fed into the classifier to identify the cytoskeleton class. Details of the class definition are described in the following section. Linear neurons as the hidden layer, and softmax to get the probability distribution. Cross entropy is taken as the loss function to update the weights and hyper-parameters in the network. The training loss, which needs to be minimized, is quantified by cross entropy as,

$$H(p, q) = -\sum_{x\in X} p(x)\log q(x), \qquad (1)$$

where $q(x)$ is the predicted probability distribution, $p(x)$ is the true probability distribution (i.e., true label).

3. EXPERIMENTAL DETAILS

To obtain actin cytoskeleton images with different morphologies, the NIH/3T3 cells were treated with Latrunculin B, a commonly known actin depolymerizer, with different concentrations. Specifically, to get enough data set, 2036, 1555, and 1674 fluorescent images of the actin cytoskeleton of NIH/3T3 cells were taken under Latrunculin B treatment of 0 nM, 40 nM, and 100 nM, respectively. These three concentrations correspond to 0%, 62%, and 100% of actin cytoskeleton depolymerization, respectively, according to the data reported (Liu et al., 2020). Thus, the actin cytoskeleton classes were defined as: Class 1 represents the original untreated status (i.e., control), Class 2 represents the 62% (relative to the untreated case) depolymerized actin cytoskeleton (40nM treatment), and

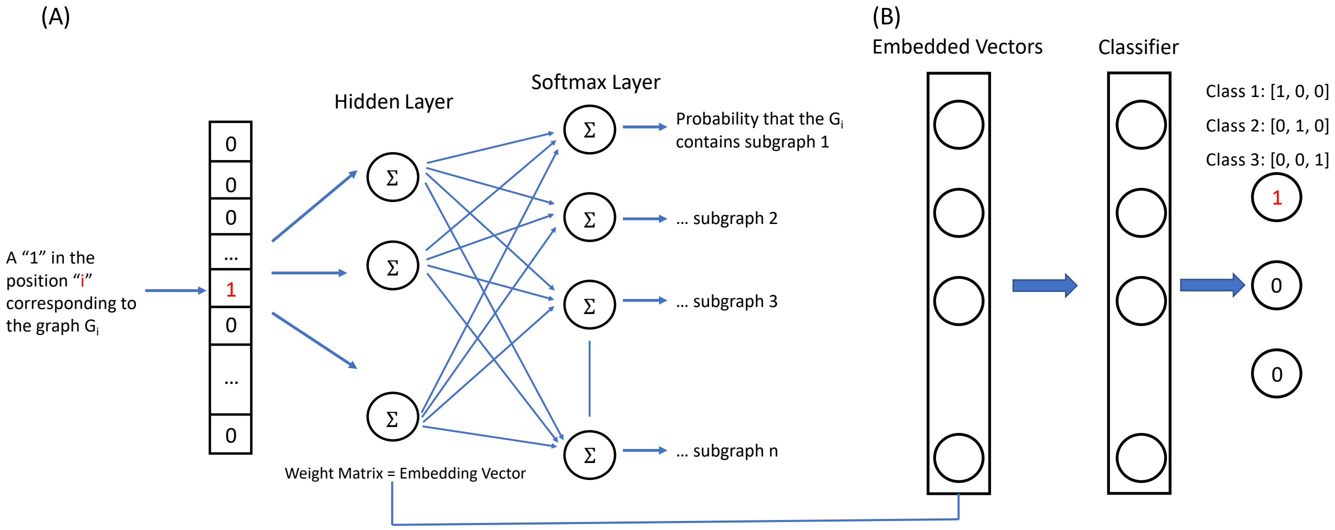

Fig. 3. The architecture of the Learning model consists of (A) an embedding layer based on the skip-gram model, and (B) a classifier based on a fully connected neural network. The given graph $G_i \in \mathbb{G}$ is fed to a hidden layer as a one-hot vector, then be passed to a softmax layer to make a prediction of subgraphs contained in the graph G_i. The embedded vectors are fed to the following classifier to do the actin cytoskeleton classification.

Class 3 represents the 100% (relative to the untreated case) depolymerized actin cytoskeleton (100 nM treatment) Total of 5265 images were preprocessed as aforementioned to be converted to actin cytoskeleton graphs.

For the classifier, the input layer has 128 neurons which is equal to the vector dimension. Three fully connected hidden layers with neuron numbers of 256, 512, 256, respectively, were developed, rectified linear unit (ReLU) was chosen as the activation for each neuron. The output layer has 3 neurons together with the softmax function as its logistic function. The output is the probability distribution of 3 classes. To train, validate (tune the hyperparameters of the trained model), and test (evaluate the accuracy) the proposed framework, the total images were randomly split into three portions (i.e., the training, the validation, and the test datasets) at fraction of 0.8, 0.1, and 0.1, respectively, with no overlap among the three sets. As a result, the size of the training dataset was 4212, the validation dataset was 527, and the test dataset was 526. To get better fitting results and training process visualization, 64 batch size and 128 epoch were applied to train the classification model. Therefore, 12,000 steps were operated in total. For every 500 steps, the training loss, training accuracy, and validation accuracy were recorded to help with the tune of hyper-parameters in the classification model.

3.1 Cell Preparation

Cell culture. Primary mouse embryonic fibroblast cells (i.e., NIH/3T3 cell) were maintained at 37°C in an incubator with the humidified atmosphere of 5% CO_2 in Dulbecco's Modified Eagle's Medium (ATCC, Rockville, MD, USA), and cultured in the medium mixed with 10%(V/V) calf bovine serum (Sigma-Aldrich, ST. Louis, MO, USA) and 1%(V/V) penicillin-streptomycin (Gibco, Grand Island, NY, USA). NIH/3T3 cells were seeded in 35 mm tissue culture dishes (Azzota Scientific, DE, USA), and then treated after 24 hours.

Actin cytoskeleton treatment. To investigate the actin cytoskeleton in different morphology states, the cells were treated with latrunculin B (Millipore Sigma, Billerica, MA, USA) at the final concentration of 0 nM, 40 nM, and 100 nM in the aforementioned cell culture medium and incubated for 30 mins. The stock solution was made by dissolving latrunculin B in Dimethyl sulfoxide (Sigma Aldrich, St. Louis, MO, USA).

Actin cytoskeleton staining. To observe the actin cytoskeleton, we fixed the cells for 10 mins using 4% paraformaldehyde (Alfa Aesar, Ward Hill, MA, USA) in PBS, and permeabilize them for 10 mins using 0.1% Triton-X (Fisher Scientific, Fair Lawn, NJ, USA) at room temperature. The actin cytoskeleton was then stained with Actin-stain^{T}M 488 phalloidin (Cytoskeleton Inc, Denver, CO, USA) at the concentration of 100 nM in PBS and kept in the dark at room temperature for 30 mins.

3.2 Fluorescence Microscope

The fluorescent actin cytoskeleton was imaged using an inverted optical microscope (IX73, Olympus, Japan) integrated with a sola light engine (Lumencor, Beaverton, OR, USA) which offers access to solid-state illumination.

4. RESULT AND DISCUSSION

4.1 Graph to Vector Embedding

The obtained graphs were embedded to vectors through the skip-gram model. Graph $G_i \in \mathbb{G}$ was fed as one-hot vectors to the embedding model. In this phase, the skip-gram model was trained to maximize the probability of predicting subgraphs that exist in the graph G_i. The number of input neurons was equal to the size of graph set $\mathbb{G}$, the number of embedding neuron was equal to the selected embedding dimension, and the number of output neurons was equal to the size of subgraph set SG_{vocab}. The shape of the weight matrix was (R, C), where $R = 5265$ is

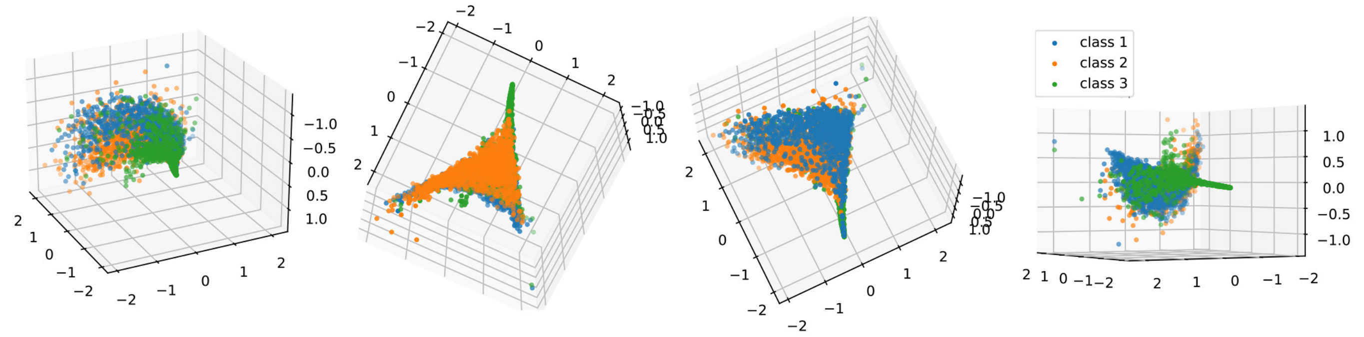

Fig. 4. Projecting embedded vectors to 3D space using PCA in four directions.

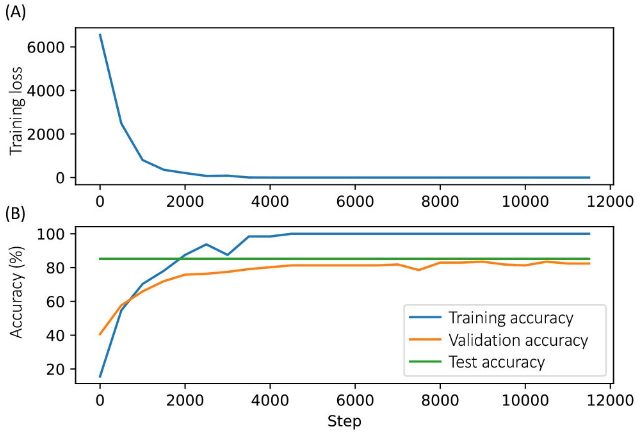

Fig. 5. (A) Recorded training loss. The blue line shows the training loss decreased from 6856 to 0. (B) Recorded accuracy during the optimization and testing process. The blue line indicates the training accuracy which starts from 16.5% to 100%. The orange line indicates the validation accuracy improved from 41.2% to 84.6%. The green line shows the final test accuracy of 85.3% after the model testing.

the number of input graphs, and $C = 128$ is the selected vector dimension. The embedded vectors are shown in Table. 1.

Table 1. Embedded vectors

No.	Class Label	x_0	x_1	...	x_{127}
1	1	-0.6151	0.9173	-0.41723	-0.1682
2	1	1.4089	-0.4357	1.8118	-0.1492
3	2	0.2851	0.0359	0.8886	-0.3929
...	...	...	...	...	...
5265	3	0.1143	0.2257	0.4484	0.1463

As aforementioned, the graphs were embedded into a vector space of 128 dimensions. To visualize the embedding result, 3-dimensional principal component analysis (3D-PCA) was applied for increasing the interpretability and minimizing the information loss at the same time. As shown in Fig. 4, the 3D distribution of embedded vectors indicates that the vectors representing graphs extracted from actin cytoskeleton indeed have their own characteristic features under different treatment concentrations. The explained variance is 97.96%, which indicates a strong strength of association between the original data and the first 3 principal components. Since only 3 principal components were presented in Fig. 4, a more significant distinction could be generated in higher dimension space. Therefore, the embedding layer efficiently reduced the graphs without missing their structure and node features.

4.2 Classifier Training and Validation

The embedded vectors were fed in the classifier to fit the corresponding labels. The training loss is shown in Fig. 5 (A). As we can see, the training loss is stably decreasing from 6856 to 0 with the training step increase. Meanwhile, the training accuracy increased and reached its stable state at 100% around 5000 steps. The validation accuracy also increased with the training step, and it reached around 84% at the end, see Fig. 5 (B). The stabilities of the loss decrease and the accuracy increase indicate that the classifier worked appropriately during the optimization process. The difference between training and validation accuracy might be caused by the lack of training samples.

4.3 Model Test

As there was no overlap between validation and test datasets, the data independence could avoid the potential effect on the model optimization from the test process. In other words, the final test accuracy is independent of the training process. As we can see in Fig. 5 (B), the final test accuracy is 85.3%. The test accuracy is close to the steady-state of the validation accuracy during the training process, which shows that the developed classifier was fitted very well on its input and outputs. This also reinforces the generalization ability of the proposed model.

The final outputs of the modeling framework were the actin cytoskeleton classes. For each class, i.e., actin cytoskeleton depolymerization degree, the corresponding cell mechanical properties, such as Young's modulus, could be pre-measured experimentally, such as using Atomic Force Microscope, and then stored in the database. Such as the three classes used in this work, the corresponding Young's modulus of NIH/3T3 cells have been reported previously (Liu et al., 2020). Therefore, by using the proposed method, other researchers could obtain the cellular elasticity of NIH/3T3 cells directly based on the actin cytoskeleton classification result generated by the proposed approach. This implies the repeated operations of biomechanical experiments are no longer needed, which could significantly improve the research efficiency and productivity.

5. CONCLUSION

In this paper, a model of the cellular actin cytoskeleton morphology was built using the graph to vector embedding technique together with the neural network classification in machine learning. The proposed model consists of a skip-gram model followed by a fully connected classifier. The actin cytoskeleton morphology is modeled based on both the structure and node features extracted from the cytoskeleton images. We demonstrated the classification accuracy of the proposed framework using actin cytoskeleton images from cells treated by Latrunculin B at different concentrations. The actin cytoskeleton morphology corresponding to each treatment concentration was defined as a class (e.g., actin depolymerization level). Test accuracy of 85.3% was obtained at the end which proved the reliability of the proposed graph embedding + classifier deep learning framework to actin cytoskeleton modeling.

ACKNOWLEDGEMENTS

The NIH/3T3 cells were provided by Dr. Xuefeng Wang. The authors also thank Ruocheng Yin and Charchit Shukla for helping with the image cropping.

REFERENCES

Bao, P., Zhang, L., and Wu, X. (2005). Canny edge detection enhancement by scale multiplication. *IEEE transactions on pattern analysis and machine intelligence*, 27(9), 1485–1490.

Goldberg, Y. and Levy, O. (2014). word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Gupta, M., Sarangi, B.R., Deschamps, J., Nematbakhsh, Y., Callan-Jones, A., Margadant, F., Mège, R.M., Lim, C.T., Voituriez, R., and Ladoux, B. (2015). Adaptive rheology and ordering of cell cytoskeleton govern matrix rigidity sensing. *Nature communications*, 6, 7525.

Harris, D. and Harris, S. (2010). *Digital design and computer architecture*. Morgan Kaufmann.

Higaki, T., Kutsuna, N., Sano, T., Kondo, N., and Hasezawa, S. (2010). Quantification and cluster analysis of actin cytoskeletal structures in plant cells: role of actin bundling in stomatal movement during diurnal cycles in arabidopsis guard cells. *The Plant Journal*, 61(1), 156–165.

Jurafsky, D. and Martin, J.H. (2000). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.

Kimori, Y., Hikino, K., Nishimura, M., and Mano, S. (2016). Quantifying morphological features of actin cytoskeletal filaments in plant cells based on mathematical morphology. *Journal of theoretical biology*, 389, 123–131.

Lichtenstein, N., Geiger, B., and Kam, Z. (2003). Quantitative analysis of cytoskeletal organization by digital fluorescent microscopy. *Cytometry Part A*, 54(1), 8–18.

Liu, Y., Mollaeian, K., and Ren, J. (2018). An image recognition-based approach to actin cytoskeleton quantification. *Electronics*, 7(12), 443.

Liu, Y., Mollaeian, K., Shamim, M.H., and Ren, J. (2020). Effect of f-actin and microtubules on cellular mechanical behavior studied using atomic force microscope and an image recognition-based cytoskeleton quantification approach. *International journal of molecular sciences*, 21(2), 392.

Lu, A.X., Kraus, O.Z., Cooper, S., and Moses, A.M. (2019). Learning unsupervised feature representations for single cell microscopy images with paired cell inpainting. *PLoS computational biology*, 15(9), e1007348.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mollaeian, K., Liu, Y., Bi, S., and Ren, J. (2018). Atomic force microscopy study revealed velocity-dependence and nonlinearity of nanoscale poroelasticity of eukaryotic cells. *Journal of the mechanical behavior of biomedical materials*, 78, 65–73.

Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7), 971–987.

Osokin, A., Chessel, A., Carazo Salas, R.E., and Vaggi, F. (2017). Gans for biological image synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, 2233–2242.

Phillip, J.M., Han, K.S., Chen, W.C., Wirtz, D., and Wu, P.H. (2021). A robust unsupervised machine-learning method to quantify the morphological heterogeneity of cells and nuclei. *Nature Protocols*, 1–21.

Shervashidze, N., Schweitzer, P., Van Leeuwen, E.J., Mehlhorn, K., and Borgwardt, K.M. (2011). Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9).

Sims, J.R., Karp, S., and Ingber, D.E. (1992). Altering the cellular mechanical force balance results in integrated changes in cell, cytoskeletal and nuclear shape. *Journal of cell science*, 103(4), 1215–1222.

Wakatsuki, T., Schwab, B., Thompson, N.C., and Elson, E.L. (2001). Effects of cytochalasin d and latrunculin b on mechanical properties of cells. *Journal of cell science*, 114(5), 1025–1036.

Wang, Y., Chen, Q., and Zhang, B. (1999). Image enhancement based on equal area dualistic sub-image histogram equalization method. *IEEE Transactions on Consumer Electronics*, 45(1), 68–75.

Wu, P.H., Phillip, J.M., Khatau, S.B., Chen, W.C., Stirman, J., Rosseel, S., Tschudi, K., Van Patten, J., Wong, M., Gupta, S., et al. (2015). Evolution of cellular morpho-phenotypes in cancer metastasis. *Scientific reports*, 5(1), 1–10.

Yanardag, P. and Vishwanathan, S. (2015). Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1365–1374.