AUTOMATED VESICLE FUSION DETECTION USING CONVOLUTIONAL NEURAL NETWORKS

*Haohan Li, Zhaozheng Yin**

Department of Computer Science
Missouri University of Science and Technology
Rolla, Missouri 65401, USA

Yingke Xu†

Department of Biomedical Engineering
Zhejiang University
Hangzhou, P.R. China

ABSTRACT

Quantitative analysis of vesicle-plasma membrane fusion events in the fluorescence microscopy, has been proven to be important in the vesicle exocytosis study. In this paper, we present a framework to automatically detect fusion events. First, an iterative searching algorithm is developed to extract image patch sequences containing potential events. Then, we propose an *event image* to integrate the critical image patches of a candidate event into a single-image joint representation as the input to Convolutional Neural Networks (CNNs). According to the duration of candidate events, we design three CNN architectures to automatically learn features for the fusion event classification. Compared on 9 challenging datasets, our proposed method showed very competitive performance and outperformed two state-of-the-arts.

Index Terms— Vesicle exocytosis, fusion event identification, convolutional neural networks

1. INTRODUCTION AND RELATED WORK

Vesicle exocytosis is an essential cellular trafficking process, by which materials (e.g., transporters, receptors and proteins) are transported from one membrane-bounded organelle to another or to the plasma membrane for growth and secretion. The analysis of these processes can provide deep insights on the cellular behavior in the diseased status [1][2].

The fusion interaction between vesicles and the cell membrane, which is able to be observed by using Total Internal Reflection Fluorescence Microscopy (TIRFM)[3][4], can be represented in 2 momentous stages (Fig.1). In stage 1, vesicles are invisible in the *pre-appearance frame*, and then suddenly appear in the *first-appearance frame* as bright fluorescent circle spots. In stage 2, after halting for several frames, vesicles will either fuse on the cell membrane with a visible "halo" (full fusion events), or depart from the cell membrane with the circular shape (partial fusion events), which can be observed in the *last appearance frame*, respectively. Finally, vesicles under the full or partial fusion event will disappear in the *disappearance frame*. As the moving trajectory of a vesicle during the fusion process is almost perpendicular to the cell membrane, the vesicle fusion event projected onto the membrane surface (i.e., the image plane in TIRFM) has minute spatial displacement.

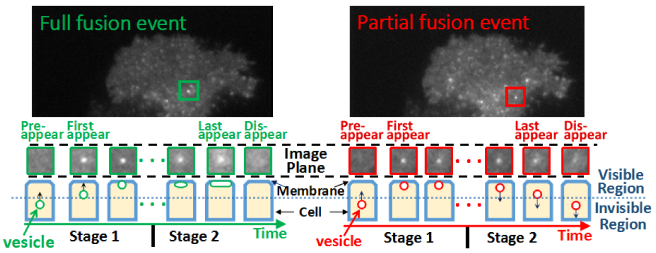

Fig. 1: The 2 momentous stages of vesicle fusions and the related 4 key frames. Here are two real TIRFM images with a full fusion event (left) and a partial fusion event (right), respectively. During the fusion events, vesicles exhibit different patterns of appearance, brightness and shape in images.

It is impractical to manually analyze TIRFM image sequences that typically consist of thousands of frames with hundreds of vesicles. Therefore, developing computational algorithms to automatically extract vesicle fusion information in TIRFM image sequences is badly needed to aid the quantitative study on the intercellular behavior.

Image processing methods have been proposed to detect fusion events [5][6][7][8]. Individual vesicles in each frame are segmented by analyzing local gray scale distributions, then full and partial fusion events are classified by a pixel intensity threshold. But these methods are sensitive to the variation of intensity profiles (shown in Fig.2(c)). Based on both temporal and spatial features, Vallotton et al. [9] proposed a filter matching method, which is able to identify the fusion events with high correlation to a standard fusion event. However, due to the frequent background intensity fluctuation (shown in Fig.2(d,e,f)) introduced by the TIRFM system and intercellular activities, it is difficult to build a template that is representative for all fusion events. In order

*This research was supported by NSF CAREER award IIS-1351049, NSF EPSCoR grant IIA-1355406, ISC and CBSE centers at Missouri S&T.

†Y. Xu was supported by the National Basic Research Program of China (2015CB352003), National Natural Science Foundation of China (31301176), Zhejiang Provincial Natural Science Foundation of China (LY13C050001) and SRFDP grant (20130101120172).

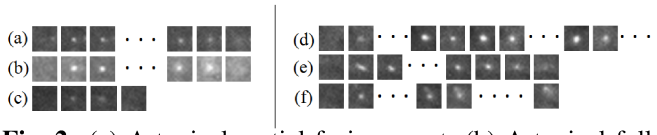

Fig. 2: (a) A typical partial fusion event; (b) A typical full fusion event; (c) A short full fusion event is characterized by its halo; (d) A bright circular object caused by the background intensity fluctuation; (e) A moving bright spot, which only moves in the first several frames then stays immobile, is similar to a partial fusion event when it stops moving; (f) A background fluctuation, which is really similar to standard full fusion event in the early stage, then gradually moves out of the field of view.

to enhance the tolerance to the variations of fusion events and the unpredictable noise interferences, some learning based methods were developed in recent years. Based on back-propagation neural network, Dosset et al. [10] developed an automatic method to detect fusion events by using a temporal sliding window. Li et al. [11] first applied a Gaussian Mixture Model (GMM) to fit on each individual fusion event, then a classifier was learned from the estimated parameters of GMMs to classify fusion events. However, the fixed temporal sliding window used in these methods may lose the critical information of fusion events with long duration.

2. CHALLENGES AND OUR PROPOSAL

Fig.2(a,b,c) show a few vesicle fusion event samples, from which we can observe some characteristics of vesicle fusion events regarding to their patterns of movement, shapes and intensities. However, it is challenging for automated image processing methods to distinguish vesicle fusion events from the large number of similar bright spots in TIRFM images. For instance, the circular background intensity fluctuation (Fig.2(d)) is similar to the vesicle fusion event. Some moving bright spots, which temporarily stay immobile near the cell membrane for several frames (Fig.2(e,f)), can be mistakenly considered as vesicle fusion events.

In this paper, we explore both appearance features and temporal cues to detect and classify fusion events. Instead of a brute-force scanning on the input image sequence to detect fusion events, we extract fusion event candidate patch sequences to improve the detection efficiency. Then, we propose to build an *event image* that mosaics the critical frames of the candidate patch sequence into a single image. In addition to the visual appearance features in individual frames, the *event image* also embeds the temporal correlation among the critical frames into a single-image joint representation, which is used as the input to Convolutional Neural Networks (CNNs) [12]. According to different lengths of the candidate patch sequences, adaptable formats of *event images* and their corresponding CNN architectures are designed to classify the candidate patch sequence into three classes: full fusion event, partial fusion event and non-fusion event.

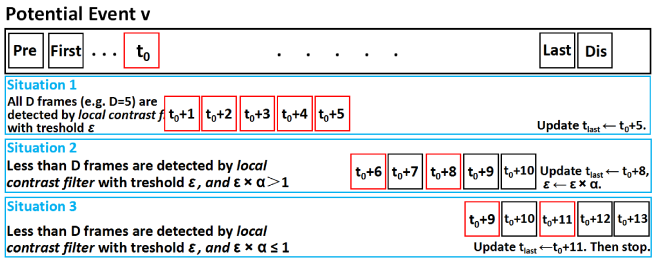

Fig. 3: An example to search the candidate patch sequence S in the forward temporal direction.

3. EXTRACT CANDIDATE PATCH SEQUENCES

As observed in the previous works [5][13][14], the vesicle fusion event appears to be a bright immobile circular spot, whose local contrast between its center and surrounding medium gradually decreases when the event disappears. Thus, we leverage the local spatial contrast to extract candidate patches in each frame, and then track them in the video sequence for the later classification, which has much better efficiency than exhaustively scanning the video volumes using spatiotemporal filters. Given the image I at time t_0, we compute the local contrast at each pixel location (x, y) as

$$f(x,y) = \frac{(n^2-1)I_{x,y}}{\sum\limits_{(i,j)} I_{i,j}} \quad (1)$$

where (i, j) represents pixels in the n-by-n neighborhood around (x, y). Pixel (x, y) is possible to belong to a fusion event if $f(x, y)$ is larger than a threshold ε. Around a potential fusion event, there might be many pixels with their local contrast larger than the threshold. We find the pixel (x^*, y^*) with the local maximum of local contrast as the center of the potential fusion event and crop an n-by-n image patch around it. Since we use fixed size patches, we only need to record the coordinates of the patch center into the fusion event candidate patch sequence, which is denoted as $S = \{x_t^*, y_t^* | t \in [t_{first}, t_{last}]\}$ where t_{first} and t_{last} denote the first and last frame index of the patch sequence, respectively. At the beginning, $t_{first} = t_{last} = t_0$.

Then, we develop an iterative searching process to find the *first-appearance frame* and the *last-appearance frame* of a potential fusion event and every patch center within this time window. We use Fig. 3 to illustrate the search in the forward direction to find the the *last-appearance frame* (the search in the backward direction to find the *first-appearance frame* is similar). During each iteration, we search the *last-appearance frame* in a sliding temporal window of D frames. Three situations are considered during the iterative search:

Situation 1, if the maximums of the local contrast in all D frames around location $(x^*_{t_{last}}, y^*_{t_{last}})$ are larger than ε, then we update $S = \{x_t^*, y_t^* | t \in [t_{first}, t_{last}]\}$ by setting $t_{last} \leftarrow t_{last} + D$ and finding the patch centers (x_t^*, y_t^*) in the D frames which are the maximums of the local contrast.

Situation 2, if not all of the maximums of the local contrast in D frames around location $(x^*_{t_{last}}, y^*_{t_{last}})$ are larger

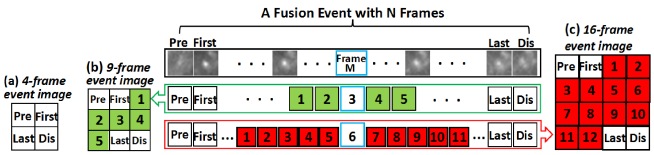

Fig. 4: Build *event image* for the CNNs.

than ε, while $\varepsilon \times \alpha > 1$ (α is a decay rate on the threshold), we update t_{last} as the last frame within the D frame whose maximal local contrast is larger than ε and the patch centers are updated accordingly. The threshold is updated as $\varepsilon \leftarrow \varepsilon \times \alpha$.

Situation 3, if not all of the maximums of the local contrast in D frames around location $(x^*_{t_{last}}, y^*_{t_{last}})$ are larger than ε and $\varepsilon \times \alpha \leq 1$, we update the patch sequence similar to situation 2, then we stop the iteration.

By applying this iterative searching algorithm to the TIRFM image sequence, we can obtain potential fusion events in the format of candidate patch sequences, each of which records the coordinates of the patch center from the *first-appearance frame* to the *last-appearance frame*. For each potential fusion event, we compute the pairwise Euclidean distance between each consecutive pair of patch centers within the candidate patch sequence. If any of these distances is larger than the neighborhood size n, this candidate patch sequence is highly possible to be a non-fusion event cause by a moving object from the background, and we remove it from the candidate list.

In the experiment, we choose the following parameter setting: neighborhood size $n = 13$, sliding temporal window length $D = 5$, the initial threshold for local contrast $\varepsilon = 1.3$ and the threshold decay rate $\alpha = 0.95$.

4. EVENT IMAGE AND CNN ARCHITECTURES

In this section, we propose an *event image* to mosaic image patches in the candidate sequence into a single image as the input to a Convolutional Neural Networks (CNNs). The *event image* contains both the visual appearance information of each individual patch and the visual correlation among different patches. The CNN automatically learns a comprehensive representation of temporal and spatial features from the *event image* for fusion event classification. By a series of parameterized layers, CNN maps each input *event image* into the probabilities of three classes: full fusion event, partial fusion event or non-fusion event.

The *event image* stitches critical patches from a candidate sequence into a single image by a specific order, which allows the CNN to discover not only the spatial and temporal information of the fusion event, but also the hidden correlation among its patches. Furthermore, we designed the *event image* as a square image so each patch has more chances to be neighbors of other patches. For example, given 16 patches, if we concatenate them into a 16-by-1 matrix pattern, there is no 4- or 8-connected neighborhood relationship among the

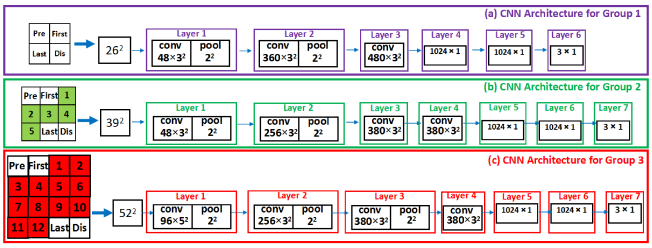

Fig. 5: Our CNN architectures. (a) The CNN architecture for vesicle fusion events in Group 1, which accepts *4-frame event images* with the size of 26×26 pixels; (b) The CNN architecture for vesicle fusion events in Group 2, which accepts *9-frame event images* with the size of 39×39 pixels; (c) The CNN architecture for vesicle fusion events in Group 3, which accepts *16-frame event images* with the size of 52×52 pixels; Note that, in all of these three architectures, each convolution process is followed by a rectified linear function (relu). Each max pooling is followed by a local normalization.

patches. Rearranging the patches into a 8-by-2 matrix pattern increases the relationship a little. If we stitch the 16 patches into a 4-by-4 matrix pattern, a lot of 4- or 8-connected neighborhood relationship can be built among the patches.

Due to the large variation of the duration of vesicle fusion events, it is unpractical to design one fixed size of *event image* that fits all vesicle fusion events well. To distinguish the *event images* containing different numbers of image patches, we name an *event image* that contains k frames as *k-frame event image* (shown in Fig.4), where k is chosen to be a squared number to insure the *event image* be square sized.

We categorize all vesicle fusion events into three groups based on their duration lengths. Group 1 contains vesicle fusion events having 4 to 6 frames, which takes image patches from the 4 key frames to construct *4-frame event images*. Group 2 contains vesicle fusion events having 9 to 13 frames, which constructs *9-frame event images*. Group 3 contains vesicle fusion events having 16 frames or more, which constructs *16-frame event images*. For the vesicle fusion event with long duration in Group 2 or Group 3, we select image patches not only from the 4 key frames that represent its appearance and disappearance moments, but also from consecutive frames around the central frame M ($M = \lceil N/2 \rceil$), which contain subtle characteristics of the variation pattern during the fusion process, as shown in Fig.4.

Then, the *event images* will be fed into the specific CNN architectures, as shown in Fig.5. In this paper, we adopt the MatConvNet [15] to design our CNN architectures. In the CNN architecture for Group 1, the first three layers are convolutional layers, where each of layer 1 and layer 2 is followed by a max-pooling that is used to extract local maximum in every 2×2 region. For the CNN for Group 2 and Group 3, we design four convolutional layers for each of them. Compared with Group 2, we design one more max-pooling following the third layer of CNNs in Group 3. In all of our CNNs, the

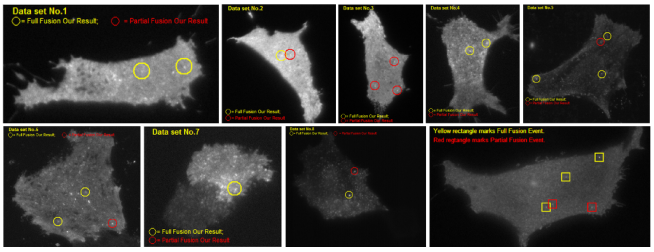

Fig. 6: Examples of our detection on 9 datasets. (yellow: full fusion; red: partial fusion)

Dataset	1	2	3	4	5	6	7	8	9
# of Frames	2663	2661	2662	2662	579	1196	1665	428	1202
# of Full Fusions	118	169	31	132	48	16	19	76	193
# of Partial Fusions	28	64	56	6	10	16	76	11	191

Table 1: The specifications of our 9 datasets.

last three layers are full connection layers. We minimize the softmax cost function at the last layer in each of these three CNNs, and use the back propagation to learn the parameters among the layers.

5. EXPERIMENTAL RESULTS

Datasets. We imaged different cell types with a variety of vesicle exocytosis in mammalian cells. These include constitutive exocytosis (transferrin receptor-pHluorin exocytosis in endothelial cells and 3T3-L1 adipocytes) and regulated exocytosis (VAMP2-pHluorin labeled insulin granule in MIN-6 cells and VAMP2-pHlurin labeled GLUT4 vesicle in 3T3-L1 adipocytes). In the experiments, 9 real TIRFM image sequences (examples are shown in Fig.6) were captured at 5 frame per second (fps), which consist of 15718 frames in total. Detailed specifications are summarized in Table 1. All datasets were well annotated by cell biologists working on vesicle trafficking analysis.

Experiment design & evaluation metric. We use the leave-one-out strategy to evaluate our method's performance, i.e., eight sequences are used for training while the last one for testing. In total, 9 leave-one-out experiments are performed on the datasets. The average performance on the 9 experiments in terms of precision, recall and F-score are used as the evaluation metrics.

Effectiveness of candidate patch sequence extraction. By using our proposed iterative searching algorithm, we obtain 4127 candidate patch sequences which contain all the 1260 vesicle fusion events (i.e., the recall is 100% and the precision is $1260/4127 = 30\%$ from the detection step). Data augmentation techniques were applied on our positive training samples to provide enough training data.

Comparison with state-of-the-arts. We compare our algorithm with two state-of-the-arts: the learning-based Gaussian Mixture Model (GMM, [11]), and the intensity-based Single Gaussian Model (SGM, [5]). All parameters in [11] and [5] are optimized to ensure they can obtain their best performance in our TIRFM image sequences for fair comparisons. As shown in Table 2, compared with the GMM [11] with handcrafted features, our method achieves much better classification results for both the full and partial fusion events in 9 datasets, which validates that the proposed *event image* and the automatic feature selection by our CNN architectures have a more comprehensive representation of vesicle fusion events. Compared with the SGM [5] that only considers the spatial radius of the Gaussian fit to the bright blob, our method outperforms it by a large margin via using both the visual features and temporal cues hidden in the *event image*.

	Full Fusion			Partial Fusion		
	Precision	Recall	F Score	Precision	Recall	F Score
Our Method	**95.0**%	**95.5**%	**95.2**%	**96.7**%	**96.1**%	**96.4**%
GMM[11]	77.0%	79.3%	78.1%	75.5%	76.0%	75.7%
SGM[5]	54.9%	64.7%	59.4%	64.6%	62.0%	63.0%
SCNN	93.7%	94.9%	94.3%	91.0%	93.2%	92.1%
MCNN	91.1%	91.0%	91.0%	88.2%	91.5%	89.8%

Table 2: The comparison of five methods on all datasets. GMM[11]: Gaussian Mixture Model; SGM[5]: Single Gaussian Model; SCNN: Single-group CNN architecture; MCNN: Multi-channel CNN architecture.

Multi-group CNN vs. Single-group CNN. We compared our multi-group CNN architectures with a Single-group CNN architecture (SCNN, i.e., for each fusion event, we only select image patches from the 4 key frames to construct the *4-frame event image* for classification). SCNN uses the architecture in Fig.5(a). As shown in Table 2, the SCNN outperformed the two state-of-the-arts, while our method using three groups of event images and CNN architectures achieves even higher performance than SCNN.

Multi-group CNN vs. Multi-channel CNN. Our proposed method is also compared with Multi-channel CNN architecture (MCNN, i.e., for every vesicle fusion event, we construct a 4-channel image by using its 4 key frames, as the input to a CNN). As shown in Table 2, both SCNN and our multi-group CNN architectures outperformed MCNN. We believe it is because the informative hidden correlation among the patches of the fusion event is incorporated into the CNN when *event images* are utilized.

6. CONCLUSION

In this paper, we first propose an iterative searching algorithm to extract patch sequences of potential fusion events, then design an *event image* to combine some informative patches of a candidate event into a single-image representation. According to different formats of *event images*, three specific Convolutional Neural Networks (CNNs) are designed to comprehensively learn the subtle characteristics of vesicle fusion events with different durations. All the potential events are classified by our CNNs into full-, partial-, or non-fusion events. Compared on 9 challenging datasets, our method showed very competitive performance and outperformed two state-of-the-arts.

7. REFERENCES

[1] S.E. Leney and J.M. Tavare, "The molecular basis of insulin-stimulated glucose uptake: signalling, trafficking and potential drug targets," *Journal of Endocrinology*, vol. 203(1), pp. 1–18, 2009.

[2] A.Bornemann, T. Ploug, and H. Schmalbruch, "Subcellular localization of GLUT4 in nonstimulated and insulin-stimulated soleus muscle of rat," *Diabetes*, vol. 41, pp. 215–221, 1992.

[3] H. Schneckenburger, "Total internal reflection fluorescence microscopy: technical innovations and novel applications," *Current Opinion in Cell Biology*, vol. 16(1), pp. 13–18, 2005.

[4] D. Axelrod, "Cell-substrate contacts illuminated by total internal reflection fluorescence," *The Journal of Cell Biology*, vol. 89, pp. 141–145, 1981.

[5] L. Bai, Y. Wang, J. Fan, et al., "Dissecting multiple steps of GLUT4 trafficking and identifying the sites of insulin action," *Cell Metabolism*, vol. 5(1), pp. 47–57, 2007.

[6] S.H. Huang, L.M. Lifshitz, C. Jones, et al., "Insulin stimulates membrane fusion and GLUT4 accumulation in clathrin coats on adipocyte plasma membranes," *Molecular and Cellular Biology*, vol. 27(9), pp. 3456–3469, 2007.

[7] A. Basset, P. Bouthemy, J. Boulanger, et al., "Localization and classification of membrane dynamics in tirf microscopy image sequences," *International Symposium on Biomedical Imaging*, 2014.

[8] A. Basset, P. Bouthemy, J. Boulanger, et al., "Detection and estimation of membrane diffusion during exocytosis in tirfm image sequences," *International Symposium on Biomedical Imaging*, 2015.

[9] P. Vallotton, D.E. James, and W.E. Hughes, "Towards fully automated identification of vesiclemembrane fusion events in TIRFM," *AIP Proceedings*, pp. 3–10, 2007.

[10] P. Dosset, P. Rassam, L. Fernandez, et al., "Automatic detection of diffusion modes within biological membranes using backpropagation neural network," *BMC Bioinformatics*, 2016.

[11] H. Li, Z. Yin, and Y. Xu, "A gaussian mixture model for automated vesicle fusion detection and classification," *Computational Methods for Molecular Imaging*, 2015.

[12] A. Krizhevsky, L. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," *Neural Information Processing Systems*, pp. 1106–1114, 2012.

[13] Y. Xu, B.R. Rubin, C.M. Orme, et al., "Dual-mode of insulin action controls GLUT4 vesicle exocytosis," *The Journal of General Physiology*, vol. 193(4), pp. 643–653, 2011.

[14] J. Wu, Y. Xu, Z. Feng, et al., "Automatically identifying fusion events between GLUT4 storage vesicles and the plasma membrane in TIRF microscopy image sequences," *Computational and Mathematical Methods in Medicine*, vol. 2015, pp. 1–7, 2015.

[15] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," *ACM Multimedia*, pp. 689–692, 2015.