Deep learning pipeline for cell edge segmentation of time-lapse live cell images

Chuangqi Wang[1*], Xitong Zhang[2*], Hee June Choi[1], Bolun Lin[2], Yudong Yu[3], Carly Whittle[1], Madison Ryan[1], Yenyu Chen[1], Kwonmoo Lee[1,4,5]

[1]Department of Biomedical Engineering
[2]Department of Computer Science
[3]Robotics Engineering Program
[4]Department of Electrical and Computer Engineering
[5]Bioinformatics and Computational Biology Program
Worcester Polytechnic Institute, Massachusetts, 01609, USA
*These authors equally contributed to this work.
Corresponding Author: Kwonmoo Lee, Email: klee@wpi.edu

Abstract

Quantitative live cell imaging has been widely used to study various dynamical processes in cell biology. Phase contrast microscopy is a popular imaging modality for live cell imaging since it does not require labeling and cause any phototoxicity to live cells. However, phase contrast images have posed significant challenges for accurate image segmentation due to complex image features. Fluorescence live cell imaging has also been used to monitor the dynamics of specific molecules in live cells. But unlike immunofluorescence imaging, fluorescence live cell images are highly prone to noise, low contrast, and uneven illumination. These often lead to erroneous cell segmentation, hindering quantitative analyses of dynamical cellular processes. Although deep learning has been successfully applied in image segmentation by automatically learning hierarchical features directly from raw data, it typically requires large datasets and high computational cost to train deep neural networks. These make it challenging to apply deep learning in routine laboratory settings. In this paper, we evaluate a deep learning-based segmentation pipeline for time-lapse live cell movies, which uses small efforts to

prepare the training set by leveraging the temporal coherence of time-lapse image sequences. We train deep neural networks using a small portion of images in the movies, and then predict cell edges for the entire image frames of the same movies. To further increase segmentation accuracy using small numbers of training frames, we integrate VGG16 pretrained model with the U-Net structure (VGG16-U-Net) for neural network training. Using live cell movies from phase contrast, Total Internal Reflection Fluorescence (TIRF), and spinning disk confocal microscopes, we demonstrate that the labeling of cell edges in small portions (5~10%) can provide enough training data for the deep learning segmentation. Particularly, VGG16-U-Net produces significantly more accurate segmentation than U-Net by increasing the recall performance. We expect that our deep learning segmentation pipeline will facilitate quantitative analyses of challenging high-resolution live cell movies.

Introduction

Time-lapse microscopy for live cell imaging allows us to access spatiotemporal information of complex cellular processes. Together with computational image analysis, it enables quantitative understanding of cellular dynamics[1-3]. Phase contrast microscopy is widely used for live cell imaging, since it does not require any labeling and rarely causes phototoxicity to live cells. Fluorescence microscopy also allows monitoring of the dynamics of fluorescently tagged molecules in live cells[4,5]. However, live cell images from these imaging modalities pose numerous challenges regarding image analysis, and conventional segmentation algorithms such as intensity thresholding[6] and Canny edge detection[7] do not produce adequate results in many cases. The challenges for cell segmentation in live cell images are as follows: i) phase contrast images contain complex image features and halo effects, which significantly reduce the reliability of conventional image analysis algorithms[8-10]. ii) fluorescence live cell images are usually noisy and low contrast since the light illumination is minimized for cell viability, and researchers need to select the cells of low level expression of fluorescent proteins, and photobleaching further degrades the image qualities, iii) strong local accumulation of fluorescence signals to subcellular structures such as focal adhesions and membrane-bound organelles, iv) uneven illumination particularly in TIRF (Total Internal Reflectance Fluorescence) microscopy make it difficult to threshold images. To resolve these issues, it is necessary to develop segmentation methods case-by-case, but they are often limited in the accurate detection of cell boundaries. Therefore, as an initial image analysis step, there is an unmet need for general, robust, and accurate cell segmentation for reliable and scalable quantitative analyses on high spatiotemporal resolution microscopy data.

The conventional methods including Otsu method[6], Canny Detector[7], active contour or snake-based method[11] and PMI method based on mutual Information[12] segment images based on the predefined local or global features. Even if these methods are routinely used for cell images, they often rely on a set of *a priori* assumptions about image characteristics, which tends to be broken in the images of complex features for segmentation. On the other hand, supervised learning methods based on deep learning achieve a higher accuracy in image segmentation since features related to edges can be learned from the

training set. Recently, deep learning (DL) has achieved great success in image classification[13-15] and segmentation[16-20], and demonstrated promising results for cell image segmentation[17,21-25]. Particularly, U-Net[17,21] can directly learn nonlinear mappings from raw images to the labeled segmented images by integrating low-level and high-level image features. However, since these methods were based on deep neural networks, it is usually necessary to prepare large training datasets. Moreover, the acquisition of high-resolution live cell movies is particularly labor-intensive since high numerical aperture objective lenses have narrow field of views, limiting data throughput. Training a deep neural network with large datasets also requires substantial computational resources and time. These make it challenging for cell biologists to take advantage of deep learning for cell segmentation in their routine laboratory settings.

In this paper, we evaluated a deep learning-based segmentation pipeline for time-lapse live cell movies, which leverages the temporal coherence of time-lapse image sequences. We trained deep neural networks using a small portion of images in the movies, and then predict cell edges for the entirety of image frames of the same movies. To test this pipeline, we employed the conventional U-Net and the transfer learning-based VGG16-U-Net, which integrates VGG16[14] pretrained model with the U-Net decoder structure. We found that live cell movies from various imaging modalities could be accurately segmented by VGG16-U-Net by labeling only 5~10% of the whole image frames, which increases the usability of deep learning in cell segmentation. This allows us to use deep learning for routine segmentation of high resolution live cell movies using low-end computational resources with a smaller training set size.

Results

Deep leaning pipelines for the segmentation of live cell movies

DL approaches to image segmentation have been focused on static cell images[21,22,26], which requires manually labeling large training sets for effective training. Building large training dataset for live cell imaging is difficult since live cell experiments usually have low data throughput. Particularly, the throughput of live cell imaging of high spatiotemporal resolution is even more limited due to the small field of view of objective lenses with high numerical aperture. Also, for accurate segmentation, high fidelity training sets are

required, substantially increasing the cost of data labeling. To overcome these challenges, we hypothesized that we can significantly reduce the burden of the training set preparation if we prepare training dataset per each movie, considering the coherence of image frames within the same time lapse movies. Therefore, we tested generating training dataset only within the frames of the total frames from a single movie provided enough data to train a deep neural network (DNN) to predict cell edges from movies with high spatiotemporal resolutions.

In this paper, we established a DL pipeline for cell segmentation of live cell movies (**Fig. 1**). For the training (**Fig. 1a**, see Methods for details), we first randomly selected a small number of training frames from time-lapse movies, and label cell edges. Then, we preprocessed the raw training images. To prepare the training set, we cropped large numbers of images patches (128X128 pixels) from the preprocessed training images. 60% of the cropped images contained cell edges (red boxes **Fig. 1a**) and 40% were randomly chosen from inside and outside of the cell (blue boxes in Image Cropping in in **Fig. 1a**). Then, we randomly split the data into training and validation sets with an 80:20 ratio, followed by the standard data augmentation. For neural network training, we employed the standard VGG16-U-Net or U-Net[17] in our DL pipeline. In VGG16-U-Net, we replaced the encoder of U-Net with VGG16 pretrained model[14]. This transfer learning approach where feature information from input data is extracted using the pre-trained networks and subsequently transfer the information to down-stream neural networks has been applied in other DL-based segmentation (FCN[16], DeepEdge[18], TernausNetV2[27]) and classification[28-35]. After the training was complete, the neural network was used for segmentation tasks (**Fig. 1b**); the entire images of movies were pre-processed and entered as input for segmentation. Finally, the edges were extracted from the predicted regions.

DL-based cell edge detection in phase contrast live cell movies.

To quantitatively assess the effectiveness of our DL pipeline, we used five phase contrast movies of migrating PtK1 cells acquired by a spinning disk confocal microscope for 200 frames at 5 sec/frame. To test our DL pipeline for these movies, we prepared the training set by manually segmenting cell edges every five frames in these movies. We trained our

pipeline using U-Net and VGG16-U-Net using the augmented datasets. To assess their performance, we trained each neural network using single movies with varying numbers of randomly selected training frames. We increased the number of training frames from 2 to 26. First, the training curve of binary cross-entropy loss function with eight training frames (**Fig. 2a**) showed that both U-Net and VGG16-U-Net had minimal overfitting. VGG16-U-Net converged faster in the training process and the final loss values are smaller than U-Net. Also, we measured the Dice coefficients, Precision, Recall, and F1-Score for each training session and calculated their mean values. The Dice coefficients demonstrated that the segmentation accuracy of VGG16-U-Net was high regardless of the number of frames used for the training in this dataset and the Dice coefficients from VGG16-U-Net were consistently higher than those of U-Net (**Fig. 2b**). We also quantify precision, recall, and F1-Score to specifically assess the accuracy of edge localization (see the method). As demonstrated in **Fig. 2c-e**, the performances of both U-Net and VGG16-U-Net increased as more training frames were added and started to saturate around 10 training frames. Although the precision values of U-Net and VGG16-U-Net are quite similar when the numbers of the training frames were more than 6 (**Fig. 2c**), VGG16-U-Net produced consistently better recall, and F1-Score than U-Net regardless of the numbers of training frames (**Fig. 2d-e**). When the number of training frames was small (2~14) (**Fig. 2f**) and high (18 ~ 26) (**Fig. 2g**), the recall and F1-Score of VGG16-U-Net were significantly higher than those of U-Net by Wilcoxon signed rank test (see the p-values in the figure legends).

We also visually confirmed that both VGG16-U-Net and U-Net produced reasonably good edge localization (**Fig. 3a-b**). Although VGG16-U-Net was better for segmenting retraction fibers than U-Net, both of them are limited in localizing the edges of thin retraction fibers accurately (Insets in **Fig. 3a-b**), consistent with the fact that the recall values were smaller than the precision (**Fig. 2c-d**). We also visualized the time evolution of edges of the entire movie frames. Both U-Net and VGG16-U-Net showed a smooth progression of edge localization. But, in comparison to U-Net, VGG16-U-Net detected much fewer floating debris. Even if one can readily remove these small objects by post-processing, it further demonstrates that VGG16-U-Net has better generalizability than U-

Net. Taken together, we demonstrated that we can use a small portion of training frames to segment all time-lapse movies if the trained model is applied within the same movies.

DL-based cell edge detection in noisy fluorescence live cell movies.

To test our DL pipeline for fluorescence microscopy, we used 13 dual-color fluorescence movies of GFP-mDia1 and SNAP-tag-TMR-actin in migrating PtK1 cells acquired by a spinning disk confocal microscope for 200 frames at 5 sec/frame. These cells expressed low levels of GFP-mDia1, which made the images highly noisy, while the high contrast images of SNAP-tag-TMR-actin were adequate for the conventional image thresholding for the segmentation for the labeling purpose (see Methods for the segmentation procedure). By thresholding actin images, we automatically labeled the edges for entire 200 image frames. The inputs of the training sets were the GFP-mDia1 image patches and the outputs were the segmented image patches from SNAP-tag-TMR-actin. The training curve of binary cross-entropy loss function demonstrated that both U-Net and VGG16-U-Net converged rapidly with minimal overfitting in the training process (**Fig. 4a**). The Dice coefficients demonstrated that the segmentation accuracy was high regardless of the number of frames used for the training in this dataset and the Dice coefficients from VGG16-U-Net were consistently higher than those of U-Net (**Fig. 4b**). In terms of specific edge localization, both U-Net and VGG16-U-Net produced high precision, recall, and F1-score when the number of training frame were more than six (3% of the entire frames) (**Fig. 4c-e**). As more training image frames were added, the recall and F1-Score slowly increased and saturated around 20 frames. Interestingly, U-Net produced slightly but significantly better precision of edge localization than VGG16-U-Net (**Fig. 4c** and **f-g**). However, the greater recall performance of VGG16-U-Net made VGG16-U-Net outperform U-Net in terms of F1-Score (**Fig. 4d-g**; Wilcoxon signed rank test, see the p-values in the figure legends).

Using this dataset, we further analyzed the roles of the VGG16 features (**Fig. 4h-j**). We used the VGG16 features only on the first-to-third (VGG16-3g) or the first-to-forth (VGG16-4g) convolution layers. We found that this partial usage of of the pretrained VGG16 features produced the results similar to VGG16-U-Net, suggesting that the low-level features of VGG16 play more important roles than the high-level features. To further

investigate the effects of training data size, we systematically increased the numbers of augmented images (**Fig. 4k-m**). We found that the size of the augmented data systematically increased the recall performance of U-Net when the number of training frames were small (2 ~ 14frames) (**Fig. 4l**). This suggests that U-Net has a limited recall ability of edge localization when the size of training set is small.

The visual inspection of predicted edges suggested that overall performance of U-Net and VGG16-U-Net was good in this noisy live cell movies (**Fig. 5a-b**). There existed some edges where the ground truth from the intensity thresholding and the prediction from U-Net and VGG16-U-Net did not match well. Consistently with the results of the phase contrast images, the retraction fibers were usually not identified by both U-Net and VGG16-U-Net (Inset 1 in **Fig. 5a**). When we overlaid the time evolution of predicted cell edges, the edges from U-Net and VGG16-U-Net produced smooth temporal edge changes (**Fig. 5c-d**).

DL-based cell edge detection in TIRF live cell movies.

Next, we tested our DL pipelines using more complex fluorescence live cell movies of paxillin-HaloTag-TMR, a marker of cell-matrix adhesions from a TIRF microscope. While these movies have higher contrast and less noise than the previous GFP-mDia1 movies, they have several technical challenges as follows: i) high intensity signals of paxillin accumulated in focal adhesions make the segmentation difficult particularly for intensity threshold-based methods, ii) the nonuniform light illumination of a TIRF microscope incurs additional issues for the segmentation, iii) the leading edge of cells could leave the thin TIRF illumination, resulting in less visible cell edges. To test our DL pipeline for these movies, we prepared the training set by manually segmenting the cell edges as we did in the phase contrast images. As shown in the loss curves in training and validation sets (**Fig. 6a**), VGG16-U-Net was able to achieve the better performance in both training and validation with much less training epochs than U-Net. Also, VGG16-U-Net showed greater dice coefficient, precision, recall and F1-Score regardless of the number of labeled image frames (**Fig. 6a-e**), and their behaviors were highly consistent with the results from the phase contrast images. When the number of training frames was small (2~14) (**Fig. 6f**) and high (18 ~ 26) (**Fig. 6g**), the recall and F1-Score of VGG16-U-Net were significantly

higher than those of U-Net by Wilcoxon signed rank test (see the p-values in the figure legends), whereas the difference of the precision were marginal or not significant.

We also visually confirmed that VGG16-U-Net produced the better results than U-Net as there are more magenta edges than cyan ones (see the inset 1 and 2 in **Fig. 7a** and inset 1 in **Fig. 7b**). In the case where the predicted edges did not match with the manually prepared ground truth (inset 2 in **Fig. 7b**), the predicted edges were usually located at the region where the intensity changes most significantly, whereas the ground truth edge can be on more faint boundaries. When we plotted time evolution of cell edges in VGG16-U-Net and U-Net, both neural networks produced smooth spatiotemporal edge changes (**Fig. 7c-d**).

Discussion

Live cell imaging became a fundamental tool to study dynamic biological processes such as cell migration, cell division, endocytosis, and organelle dynamics. Since segmentation is the initial step of image analysis, accurate and effective segmentation of live cell images is crucial particularly for edge velocity measurement[36-39]. In this paper, based on the temporal coherence of time-lapse image sequences, we established a DL pipeline for the segmentation of live cell time-lapse movies. We also demonstrated that the pretrain model-based VGG16-U-Net is more effective than U-Net, particularly for the recall performance. The pretrained models such VGG16 or VGG19 trained on natural images in ImageNet database have been widely used for transfer learning[28-35]. In this framework, we evaluated the performance of VGG16 feature extractor to various live cell microscopy images. We confirmed that the image descriptors from VGG16 were highly effective in predicting accurate cell edges, and required less epochs for training.

Building up the training set across various movies with different cell types, molecules, and experimental/imaging conditions will require tremendous human labors, which is difficult for small individual laboratories. In this paper, we demonstrated that building the neural network models for specific time-lapse movies produces accurate edge localization with

much less training sets and computational resources than expected, particularly when we use a transfer learning approach. We expect that this approach may not be limited to time lapse movies. For example, the cell images from the same batch with the same imaging conditions may forgo the requirement of large training sets. Therefore, we expect that our DL pipeline may accelerate the adoption of deep learning techniques by the cell biology community.

Methods

Data Collection. Cell culture and live cell imaging procedures were followed according to previous studies[36]. PtK1 cells were cultured in Ham's F12 medium (Invitrogen) supplemented with 10% FBS, 0.1 mg ml^{-1} streptomycin, and 100 U ml^{-1} penicillin. Cells were then imaged at 5 s intervals for 1000 s using 0.45 NA Super Plan Fluor ELWD 20XC ADM objective for phase contrast imaging and 60X, 1.4 NA Plan Apochromat objective for fluorescence spinning disk confocal imaging, 1.49NA Apochromat TIRF 100XC for fluorescence TIRF imaging.

PtK1 cells were transfected with the DNA constructs of GFP-mDia1 and SNAP-tag-actin or paxilin-HaloTag using Neon transfection system (Invitrogen) according to the manufacturer's instructions (1 pulse, 1400 V, 20 ms) and were grown on acid-washed glass #1.5 coverslips for 2 days before imaging. Prior to imaging, expressed SNAP-tag-actin or paxilin-HaloTag proteins were labeled with SNAP-tag-TMR (New England BioLabs) or HaloTag-TMR (Promega) ligands respectively according to the manufacturers' instructions. All imaging was performed in imaging medium (Leibovitz's L-15 without phenol red, Invitrogen) supplemented with 10% fetal bovine serum (FBS), 0.1 mg ml^{-1} streptomycin, 100 U ml^{-1} penicillin, 0.45% glucose, 1.0 U ml^{-1} Oxyrase (Oxyrase Inc.) and 10 mM Lactate. Cells were then imaged at 5 s intervals for 1000 s. PtK1 cells were acquired from Gaudenz Danuser lab. They were routinely tested for mycoplasma contamination.

All microscopy except for TIRF microscopy (described elsewhere[38]) was performed using the set up as follows: Nikon Ti-E inverted motorized microscope (including motorized focus, objective nosepiece, fluorescence filter turret, and condenser turret) with integrated Perfect Focus System, Yokogawa CSU-X1 spinning disk confocal head with manual

emission filter wheel with Spectral Applied Research Borealis modification, Spectral Applied Research custom laser merge module (LMM-7) with AOTF and solid state 445 nm (200 mW), 488 nm (200 mW), 514 nm (150 mW), 561 nm (200 mW), and 637 nm (140 mW) lasers, Semrock 405/488/561/647 and 442/514/647 dichroic mirrors, Ludl encoded XY stage, Ludl piezo Z sample holder for high speed optical sectioning, Prior fast transmitted and epi-fluorescence light path shutters, Hamamatsu Flash 4.0 LT sCMOS camera, 37 °C microscope incubator enclosure with 5% CO2 delivery (In Vivo), Molecular Devices MetaMorph v7.7, TMC vibration-isolation table.

Data Labeling. We collected five videos of PtK1 cells from a phase contrast microsope and seven videos of PtK1 cells expressing paxilin-HaloTag from a TIRF microscope. Each video contained 200 frames. We manually labeled the cell boundary every five frames based on the experiences of phase contrast and TIRF miscroscopy, which provided us with 40 labeled frames for each video for training and testing. We also collected 13 dual-color videos of PtK1 cells expressing GFP-mDia1 and SNAP-tag-actin. Since actin images had good contrast along the cell boundary, we applied intensity thresholding to prepare the ground truth segmentation for GFP-mDia1 video. For each actin image, we applied Non-local Means method implemented in ImageJ for denoising (sigma = 15 and smoothing_factor =1). Then, we manually selected the optimal threshold to segment all the frames. After that, we visually checked all masks and re-adjusted the threshold for better segmentation. The resulting binary masks were manually used as the ground-truth for GFP-mDia1 fluorescence videos.

Pre-processing. Fluorescence live images usually have poor constrast in comparison to phase contrast images. Therefore, in order to provide images with better contrast with neural network training, raw fluorescence images were pre-processed as follows (no preprocessing was applied to phase contrast images):

1) For each video, all the pixel values were collected from the labelled frames. Then, we calculated the mean μ and the standard deviation δ of pixel values.

2) We replaced the pixel values $x_{i,j}$ with the follows values when they are less than $\mu - 2\delta$ or greater than $\mu + 3\delta$.

$$x_{i,j} = \begin{cases} \mu - 2\delta, & x_{i,j} < \mu - 2\delta \\ \mu + 3\delta, & x_{i,j} \geq \mu + 3\delta \end{cases}$$

3) We applied the min-max normalization to rescale the pixel ranges to [0, 255].

$$y_{i,j} = 255 \frac{x_{i,j} - \min(x_{i,j})}{\max(x_{i,j}) - \min(x_{i,j})}$$

Training Dataset Preparation. We randomly selected the specified number of training frames from a live cell movie and the corresponding labeled segmented images. Then, we randomly cropped 200 patches (128X128 pixels) from each frame. The 60% of the cropped patches contains the edge boundary and 40% constains only foreground or background. The mean and standard deviation of collected image patches are calculated for further normalization.

After preprocessing, the standard data augmentation process was applied using the *ImageDataGenerator* implemented in Keras package. The parameters in the *ImageDataGenerator* for the data augmentation is as follows: *rotation_range=50., width_shift_range=0.1, height_shift_range=0.1, shear_range=0.1, zoom_range=0.1, horizontal_flip=True, vertical_flip=True, fill_mode='reflect'*.

The default number of augmented images was 6400 (50×128). When we evaluated the model performance by the number of augmented data, we varied the numbers from 1280 (10×128) to 8960 (70×128). After the augmentation, we pooled the cropped samples and augmented samples and randomly splitted them into the training (80%) and (20%) validation sets.

VGG16-U-Net Architecture. The VGG16-U-Net consists of an encoder and a decoder. In the encoder, the same structure of VGG-16 is applied, which contains five convolutional layers, each of which contains a different number of convolution and max-pooling operations with the depth of 64-128-256-512-512. The weights of the encoder are transferred and fixed from the VGG16 trained with ImageNet database. In the decoder, four deconvolution layers by up-sampling operations with the depth of 512-256-128-64-1 integrate the edge information directly extracted from the image as a lower level and reconstruction from region information as a higher level, as suggested in the original U-

Net. Our framework takes advantage of the VGG16 pretrained model to extract useful features. In the convolution operation, the zero-padding strategy is applied to make the same size after convolution for convenience. The size of input patch is 128x128. The size of the convolutional filter is 3x3, and the size of max-pooling is 2x2 while that of up-pooling is 2x2 to fit the size of convolution. In order to eliminate the boundary effects of zero-padding, we cropped the central parts of output segmented images with the size of 68x68.

In the prediction step, since U-Net and VGG16-U-Net are fully convolutional, we used the entire image without cropping as inputs to obtain predicted segmentation images. This prediction using entire images eliminates the boundary effects due to image cropping. The entire images were padded with 30-pixel width and height for inputs using the *copyMakeBorder* function in OpenCV package and then the padded regions were removed from the predicted output images.

Neural Network Training. The binary cross-entropy was used as a loss function for training. Adam was used as an optimizer, and the initial learning rate was 10^{-5}, and other parameters were default values in the Keras. To avoid overfitting, we used the early stopping. We stopped the training when the validation loss did not decrease 0.0001 in consecutive three epochs. The maximum epoch was 30, and the batch size was 64. When the training process was finished, the model weights were saved for further analysis. The neural network training was performed using Keras with TensorFlow backend on NVIDIA GTX 1080Ti or Titan X.

Performance evaluation of edge localization. We used the labelled data which are not included in the training process (training/validation sets) for the performance evaluation. We generated the binarized mask images by threshoding the softmax output images using the *im2bw* (MATLAB) function with the threshold value of 0.5. After we filled the small holes in the binarized mask using the *imfill* (MATLAB) function, we extracted the regions with the maximum area. Then, we generated edge images using the *bwboundaries* (MATLAB) function. The edge images were predicted for all the image frames including the labeled and unlabeled images. We used the predictions of labeled images not used for training to evaluate the performance of edge localization and those of whole image frames for visualization of edge progression.

We calculated Dice coefficients to evaluate the segmentation performance along the edge boundary as follows. First, we dilated the labeled edge masks with the kernel size 64 to generate the dilated edge masks using *dilate* function in OpenCV. Then, we applied binary AND operation between the dilated edge masks and the masks of the ground truth and the predicted masks from U-Net and VGG16-U-Net using the *bitwise_and* function in OpenCV package. After that, we calculate the Dice coefficient between the ground truth and the predictions for each image frame using the following formula ($|A \cap B|$: the common elements between sets A and B, $|A|$ and $|B|$: the number of elements in set A and B).

$$Dice(A, B) = \frac{2|A \cap B|}{|A| + |B|}$$

To specifically assess the accuracy of edge localization, we matched DL-generated edges and labeled edges by bipartite matching using the customed package implemented in Berkeley Segmentation Benchmark[40,41], with the search radii (Phase Contrast: 13 pixels; Confocal 10 pixels, TIRF: 5 pixels). The DL generated edge pixels that do not match with labelled edges are considered as false positives. Using this information, we calculate precision, $tp/(tp + fp)$ and recall, $tp/(tp + fn)$ to specifically assess the accuracy of edge localization (tp: # of true positives, fp: # of false positives, fn: # of false negatives), and estimate an F1-Score in each frame. The statistical testing of ther performance measures was performed by Wilcoxon signed rank test.

Code availability statement

The code used in the current study is available from the corresponding author upon reasonable request.

Data availability statement

The datasets used in the current study are available from the corresponding author on reasonable request.

Acknowledgements

We thank NVIDIA for providing us with TITAN X GPU cards (NVIDIA Hardware Grant Program), Microsoft for providing us with Azure cloud computing resources (Microsoft Azure Research Award), and Boston Scientific for providing us with the gift for deep learning research. This work was supported by the WPI Start-up Fund for new faculty and NIH grant GM122012.

Author Contributions

C. W. and X. Z. initiated the project, designed the algorithm, trained the neural networks and wrote the final version of the manuscript and supplement. H. C. performed the live cell imaging experiments and prepared the training sets. B. L. and Y. Y. wrote the code and trained the neural networks. C. W. M. R., and Y. C. prepared the training sets. K. L. coordinated the study and wrote the final version of the manuscript and supplement. All authors discussed the results of the study.

Competing Interests

The authors declare no competing financial or non-financial interests.

Author Information

Correspondence and requests for materials, data, and code should be addressed to K.L. (klee@wpi.edu).

References

1 Ettinger, A. & Wittmann, T. Fluorescence live cell imaging. *Methods in cell biology* **123**, 77 (2014).

2 Frigault, M. M., Lacoste, J., Swift, J. L. & Brown, C. M. Live-cell microscopy–tips and tools. *J Cell Sci* **122**, 753-767 (2009).

3 Waters, J. C. Live-cell fluorescence imaging. *Methods in cell biology* **114**, 125-150 (2013).

4 Wu, K., Gauthier, D. & Levine, M. D. Live cell image segmentation. *IEEE Transactions on biomedical engineering* **42**, 1-12 (1995).

5 Stephens, D. J. & Allan, V. J. Light microscopy techniques for live cell imaging. *Science* **300**, 82-86 (2003).

6 Otsu, N. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics* **9**, 62-66 (1979).

7 Canny, J. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, 679-698 (1986).

8 Li, K. & Kanade, T. Nonnegative mixed-norm preconditioning for microscopy image segmentation. *International Conference on Information Processing in Medical Imaging*, 362-373 (2009).

9 Ambühl, M. E., Brepsant, C., Meister, J. J., Verkhovsky, A. B. & Sbalzarini, I. F. High-resolution cell outline segmentation and tracking from phase-contrast microscopy images. *Journal of microscopy* **245**, 161-170 (2012).

10 Bensch, R. & Ronneberger, O. Cell segmentation and tracking in phase contrast images using graph cut with asymmetric boundary costs. *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, 1220-1223 (2015).

11 Chan, T. F. & Vese, L. A. Active contours without edges. *IEEE Transactions on image processing* **10**, 266-277 (2001).

12 Isola, P., Zoran, D., Krishnan, D. & Adelson, E. H. Crisp boundary detection using pointwise mutual information. *European Conference on Computer Vision*, 799-814 (2014).

13 Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 1097-1105 (2012).

14 Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations* (2015).

15 He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778 (2016).

16 Long, J., Shelhamer, E. & Darrell, T. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431-3440 (2015).

17 Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234-241 (2015).

18 Bertasius, G., Shi, J. & Torresani, L. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4380-4389 (2015).

19 Shen, W., Wang, X., Wang, Y., Bai, X. & Zhang, Z. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3982-3991 (2015).

20 Badrinarayanan, V., Handa, A. & Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293* (2015).

21 Falk, T. *et al.* U-Net: deep learning for cell counting, detection, and morphometry. *Nat Methods* **16**, 67-70, doi:10.1038/s41592-018-0261-2 (2019).

22 Sadanandan, S. K., Ranefall, P., Le Guyader, S. & Wahlby, C. Automated Training of Deep Convolutional Neural Networks for Cell Segmentation. *Sci Rep* **7**, 7860, doi:10.1038/s41598-017-07599-6 (2017).

23 McQuin, C. *et al.* CellProfiler 3.0: Next-generation image processing for biology. *PLoS Biol* **16**, e2005970, doi:10.1371/journal.pbio.2005970 (2018).

24 Moen, E. *et al.* Deep learning for cellular image analysis. *Nat Methods*, doi:10.1038/s41592-019-0403-1 (2019).

25 Chai, X., Ba, Q. & Yang, G. Characterizing Robustness and Sensitivity of Convolutional Neural Networks in Segmentation of Fluorescence Microscopy Images. *2018 25th IEEE International Conference on Image Processing (ICIP)*, 3838-3842 (2018).

26 Van Valen, D. A. *et al.* Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. *PLoS computational biology* **12**, e1005177 (2016).

27 Iglovikov, V., Seferbekov, S. S., Buslaev, A. & Shvets, A. TernausNetV2: Fully Convolutional Network for Instance Segmentation. *CVPR Workshops*, 233-237 (2018).

28 Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 3320-3328 (2014).

29 Razavian, A. S., Azizpour, H., Sullivan, J. & Carlsson, S. CNN features off-the-shelf: an astounding baseline for recognition. *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, 512-519 (2014).

30 Donahue, J. *et al.* Decaf: A deep convolutional activation feature for generic visual recognition. *International conference on machine learning*, 647-655 (2014).

31 Oquab, M., Bottou, L., Laptev, I. & Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 1717-1724 (2014).

32 Zeiler, M. D. & Fergus, R. Visualizing and understanding convolutional networks. *European conference on computer vision*, 818-833 (2014).

33 Choi, J. Y. *et al.* Multi-categorical deep learning neural network to classify retinal images: A pilot study employing small database. *PLoS One* **12**, e0187336, doi:10.1371/journal.pone.0187336 (2017).

34 Kim, S. J. *et al.* Deep transfer learning-based hologram classification for molecular diagnostics. *Sci Rep* **8**, 17003, doi:10.1038/s41598-018-35274-x (2018).

35 Pratt, L. Y. Discriminability-based transfer between neural networks. *Advances in neural information processing systems*, 204-211 (1993).

36 Wang, C. *et al.* Deconvolution of subcellular protrusion heterogeneity and the underlying actin regulator dynamics from live cell imaging. *Nat Commun* **9**, 1688, doi:10.1038/s41467-018-04030-0 (2018).

37 Machacek, M. *et al.* Coordination of Rho GTPase activities during cell protrusion. *Nature* **461**, 99-103, doi:10.1038/nature08242 (2009).

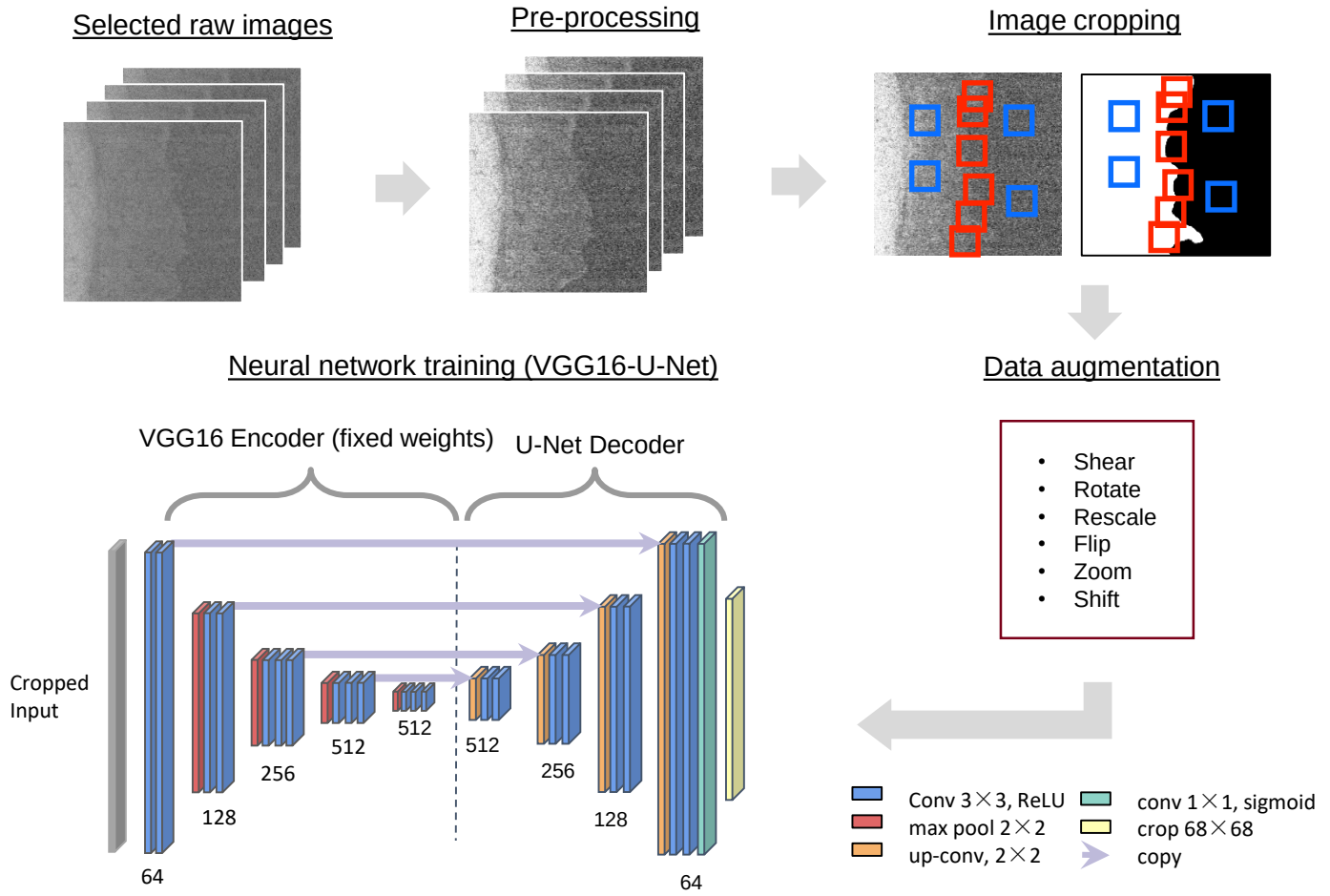
38 Lee, K. *et al.* Functional hierarchy of redundant actin assembly factors revealed by fine-grained registration of intrinsic image fluctuations. *Cell Syst* **1**, 37-50, doi:10.1016/j.cels.2015.07.001 (2015).

39 Machacek, M. & Danuser, G. Morphodynamic profiling of protrusion phenotypes. *Biophys J* **90**, 1439-1452, doi:10.1529/biophysj.105.070383 (2006).

40 Martin, D. R., Fowlkes, C. C. & Malik, J. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 530-549 (2004).

41 Arbelaez, P., Maire, M., Fowlkes, C. & Malik, J. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence* **33**, 898-916 (2010).

a. Training Pipeline



b. Prediction Pipeline

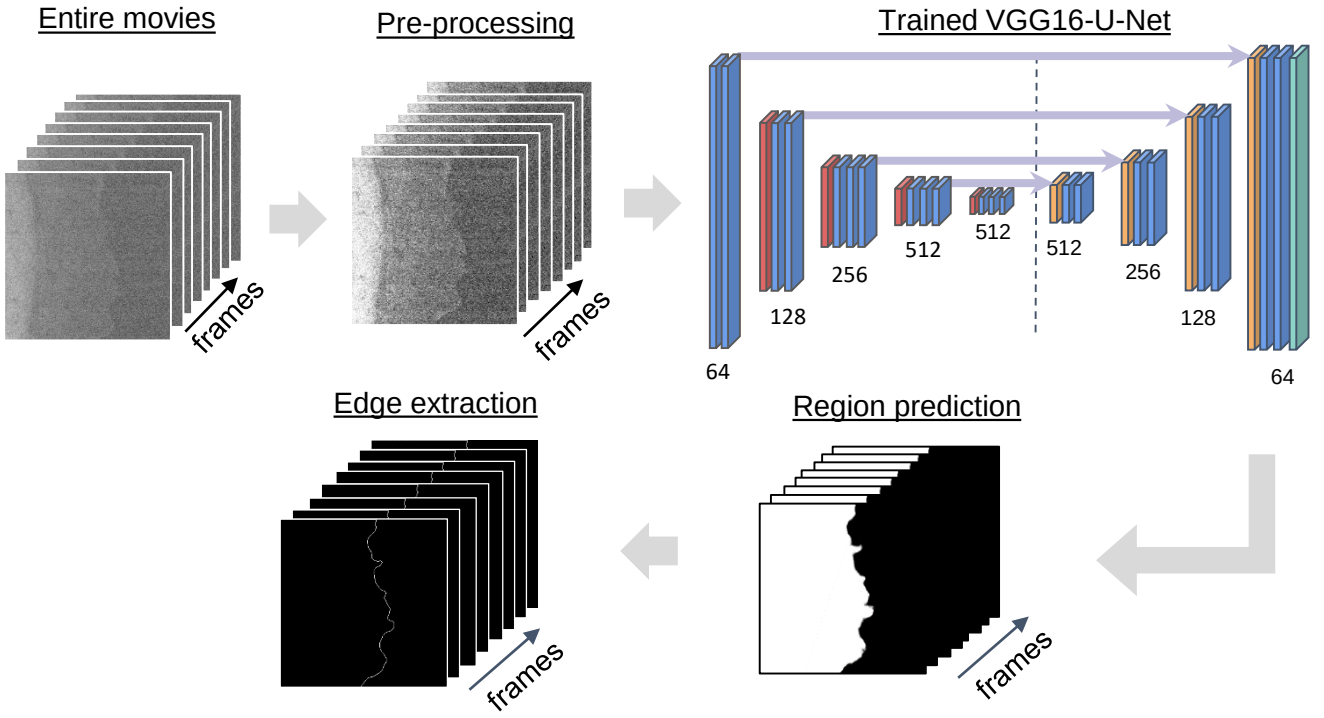


Figure 1. Deep learning pipeline of edge segmentation of live cell movies

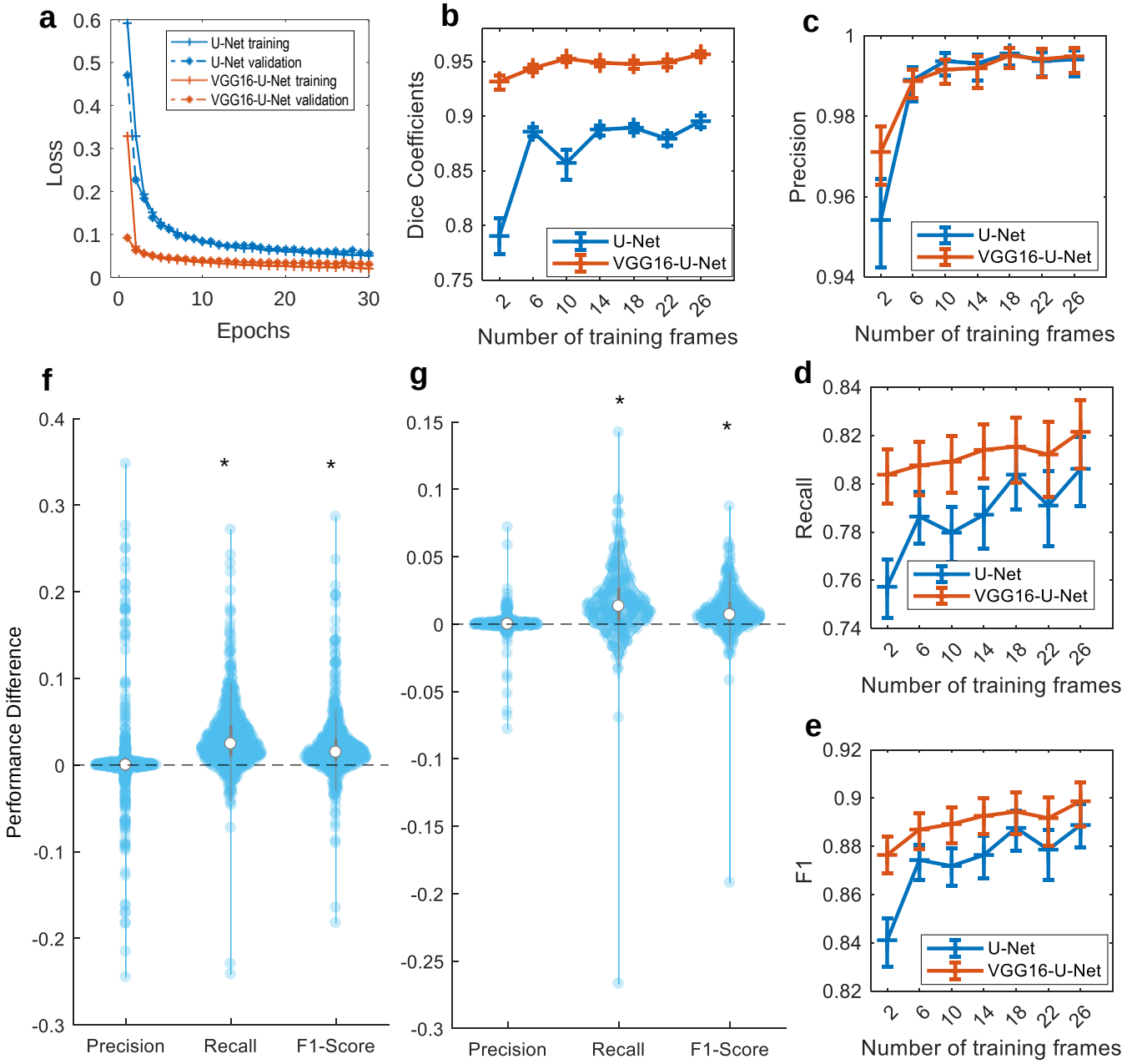


Figure 2. Segmentation performance of U-Net and VGG16-U-Net for high-resolution fluorescence movies from a phase contrast microscope. (**a**-**e**) Comparison between U-Net and VGG16-U-Net for training curves (a), dice coefficients (b), precision (c), recall (d), and F1-Score (e). (**f**-**g**) Violin plots of the difference between VGG16-U-Net and U-Net in precision, recall, and F1-Score when the numbers of training frames were 2 ~14 (f) and 18 ~ 26 (g). * indicates the statistical significance between the differences of U-Net and VGG16-U-Net with p-values, 0.80 (precision in f), 4.1×10^{-83} (recall in f), 1.86×10^{-65} (F1-Score in f), 0.085 (precision in g), 6.81×10^{-27}(recall in g), and 2.6×10^{-25} (F1-Score in g) respectively by Wilcoxon signed rank test. Error bars: 95% confidence intervals of mean

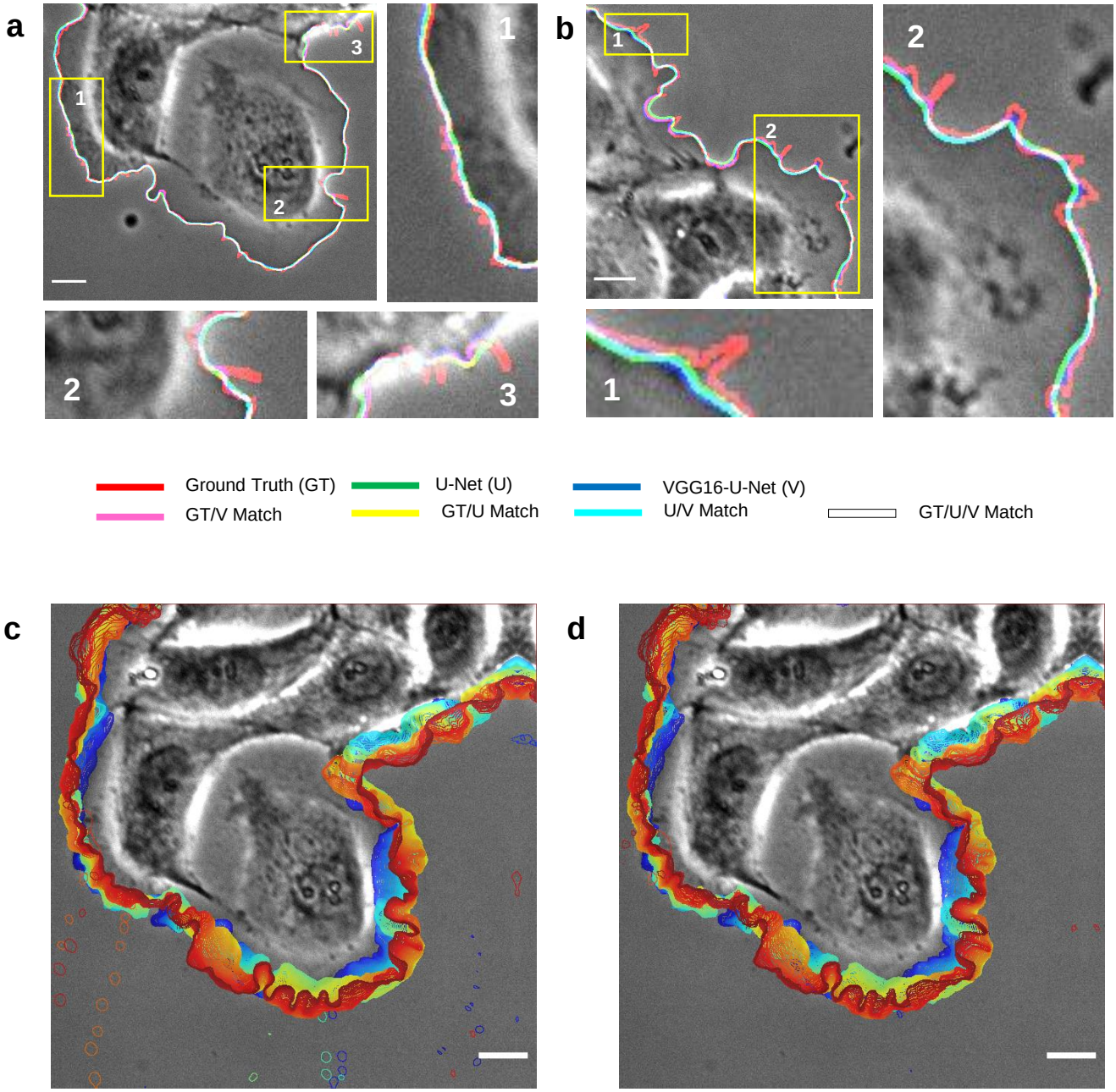


Figure 3. Visualization of segmentation results of U-Net and VGG16-U-Net for phase contrast movies of PtK1. (**a**-**b**) Examples of overlaid edges of ground truth, U-Net, and VGG16-U-Net. The number of training frames: 10. (**c**-**d**) Overlay of edges (blue, 0s; red, 1000s time points) from U-Net (c) and VGG16-U-Net (d). The number of training frames: 26. Bars: 20 μm.

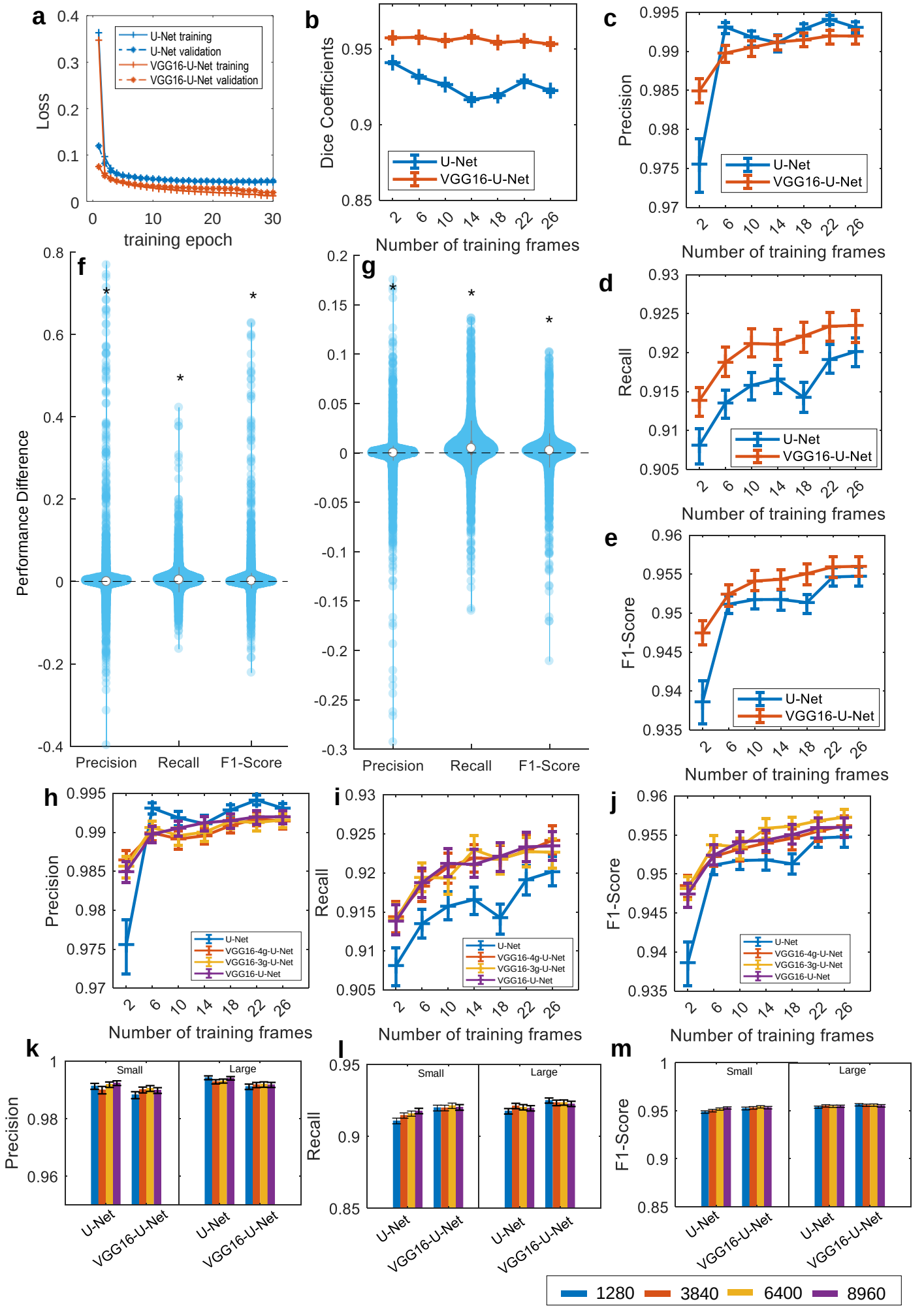

a
Loss
training epoch
U-Net training
U-Net validation
VGG16-U-Net training
VGG16-U-Net validation
b
Dice Coefficients
Number of training frames
U-Net
VGG16-U-Net
c
Precision
Number of training frames
U-Net
VGG16-U-Net
d
Recall
Number of training frames
U-Net
VGG16-U-Net
e
F1-Score
Number of training frames
U-Net
VGG16-U-Net
f
Performance Difference
Precision
Recall
F1-Score
g
Precision
Recall
F1-Score
h
Precision
Number of training frames
U-Net
VGG16-4g-U-Net
VGG16-3g-U-Net
VGG16-U-Net
i
Recall
Number of training frames
U-Net
VGG16-4g-U-Net
VGG16-3g-U-Net
VGG16-U-Net
j
F1-Score
Number of training frames
U-Net
VGG16-4g-U-Net
VGG16-3g-U-Net
VGG16-U-Net
k
Precision
Small
Large
U-Net
VGG16-U-Net
U-Net
VGG16-U-Net
l
Recall
Small
Large
U-Net
VGG16-U-Net
U-Net
VGG16-U-Net
m
F1-Score
Small
Large
U-Net
VGG16-U-Net
U-Net
VGG16-U-Net
1280
3840
6400
8960

Figure 4 Segmentation performance of U-Net and VGG16-U-Net for noisy high-resolution fluorescence movies. (**a**-**e**) Comparison between U-Net and VGG16-U-Net for training curves (a), dice coefficients (b), precision (c), recall (d), and F1-Score (e). (**f**-**g**) Violin plots of the difference between VGG16-U-Net and U-Net in precision, recall, and F1-Score when the numbers of training frames were 1 ~14 (f) and 18 ~ 26 (g). * indicates the statistical significance between the differences of U-Net and VGG16-U-Net with p-values, 3.02×10^{-21} (precision in f), 3.81×10^{-188} (recall in f), 1.87×10^{-95} (F1-Score in f), and 2.4×10^{-18} (precision in g), 1.59×10^{-193} (recall in g), 1.7×10^{-121} (F1-Score in g) respectively by Wilcoxon signed rank test. (**h**-**j**) Effects of the convolutional layers of fixed weight in VGG16 pretrained model with varying numbers of training frame on precision (h), recall (i), and F1-Score (j). The weights of the first three layers of VGG16-3g-U-Net and the first four layers of VGG16-4g-U-Net were fixed during the training. (**k**-**m**) Effects of the numbers of augmented images on precision (h), recall (i), and F1-Score (j) when the numbers of training frames were small (2 ~14; left) and large (18 ~ 26; right). Error bars: 95% confidence intervals of mean

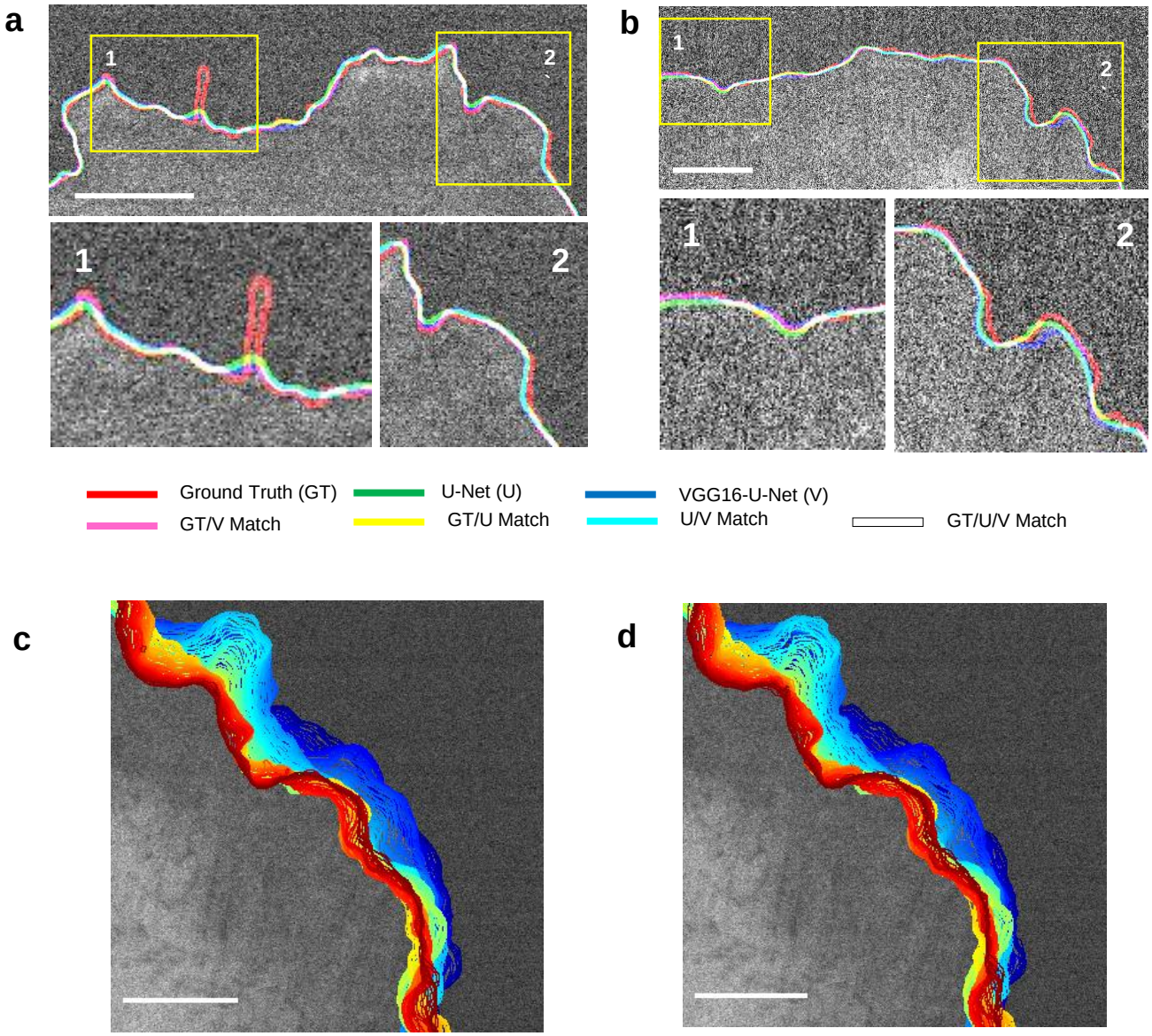


Figure 5. Visualization of segmentation results of U-Net and VGG16-U-Net for noisy high-resolution fluorescence movies of the leading edge of a PtK1 cell expressing GFP-mDia1 (the number of training frames: 26). (**a**-**b**) Examples of overlaid edges of ground truth, U-Net, and VGG16-U-Net. (**c**-**d**) Overlay of edges (blue, 0s; red, 1000s time points) from U-Net (c) and VGG16-U-Net (d). Bars: 10 μm.

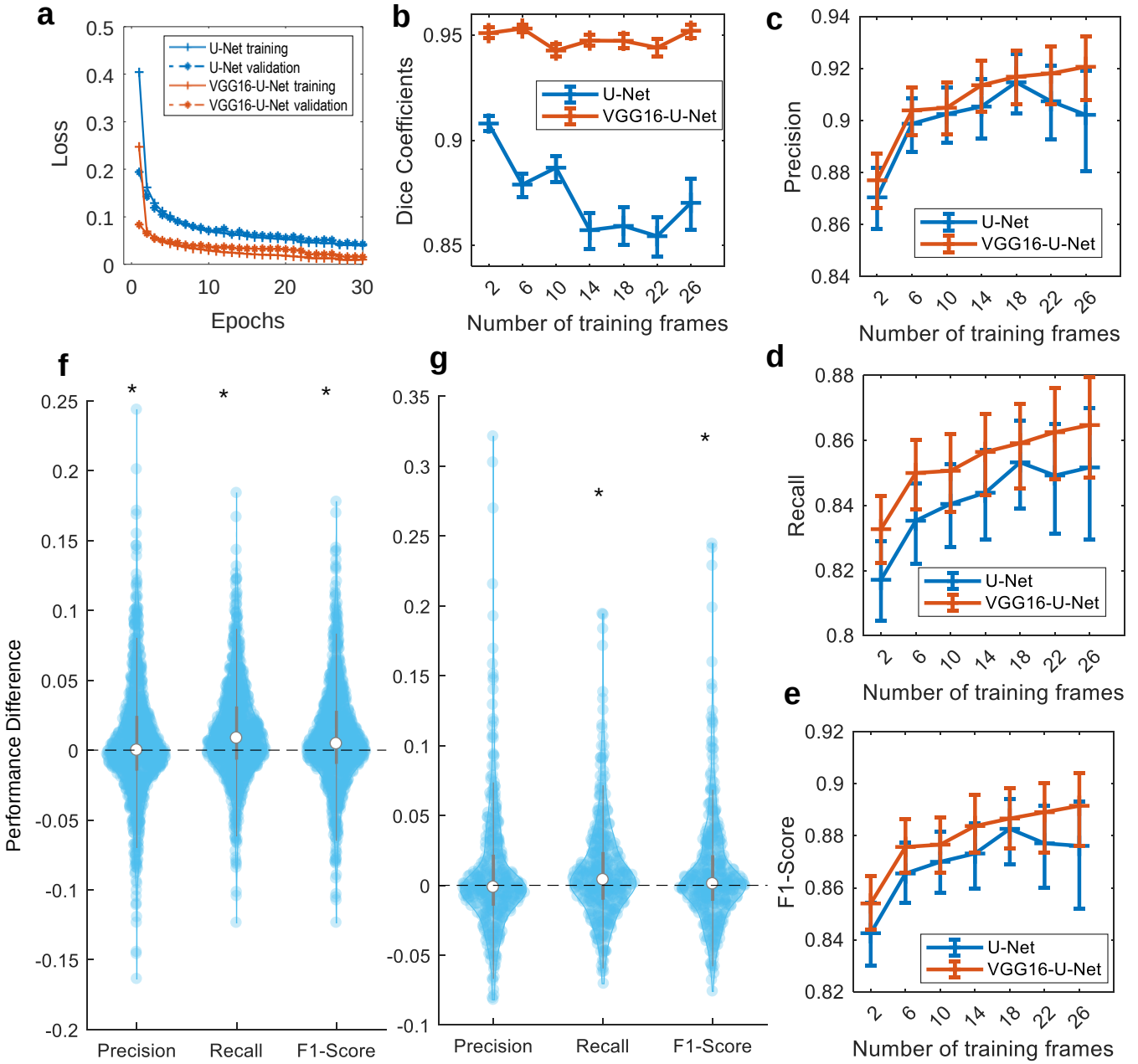


Figure 6. Segmentation performance of U-Net and VGG16-U-Net for high-resolution fluorescence movies of paxillin from a TIRF microscope. (**a-e**) Comparison between U-Net and VGG16-U-Net for training curves (a), dice coefficients (b), precision (c), recall (d), and F1-Score (e). (**f-g**) Violin plots of the difference between VGG16-U-Net and U-Net in precision, recall, and F1-Score when the numbers of training frames were 2 ~14 (f) and 18 ~ 26 (g). * indicates the statistical significance between the differences of U-Net and VGG16-U-Net with p-values, 0.0098 (precision in f), 5.98×10^{-27} (recall in f), 3.90×10^{-12} (F1-Score in f), 0.34 (precision in g), 2.21×10^{-5} (recall in g), and 0.0063 (F1-Score in g) respectively by Wilcoxon signed rank test. Error bars: 95% confidence intervals of mean

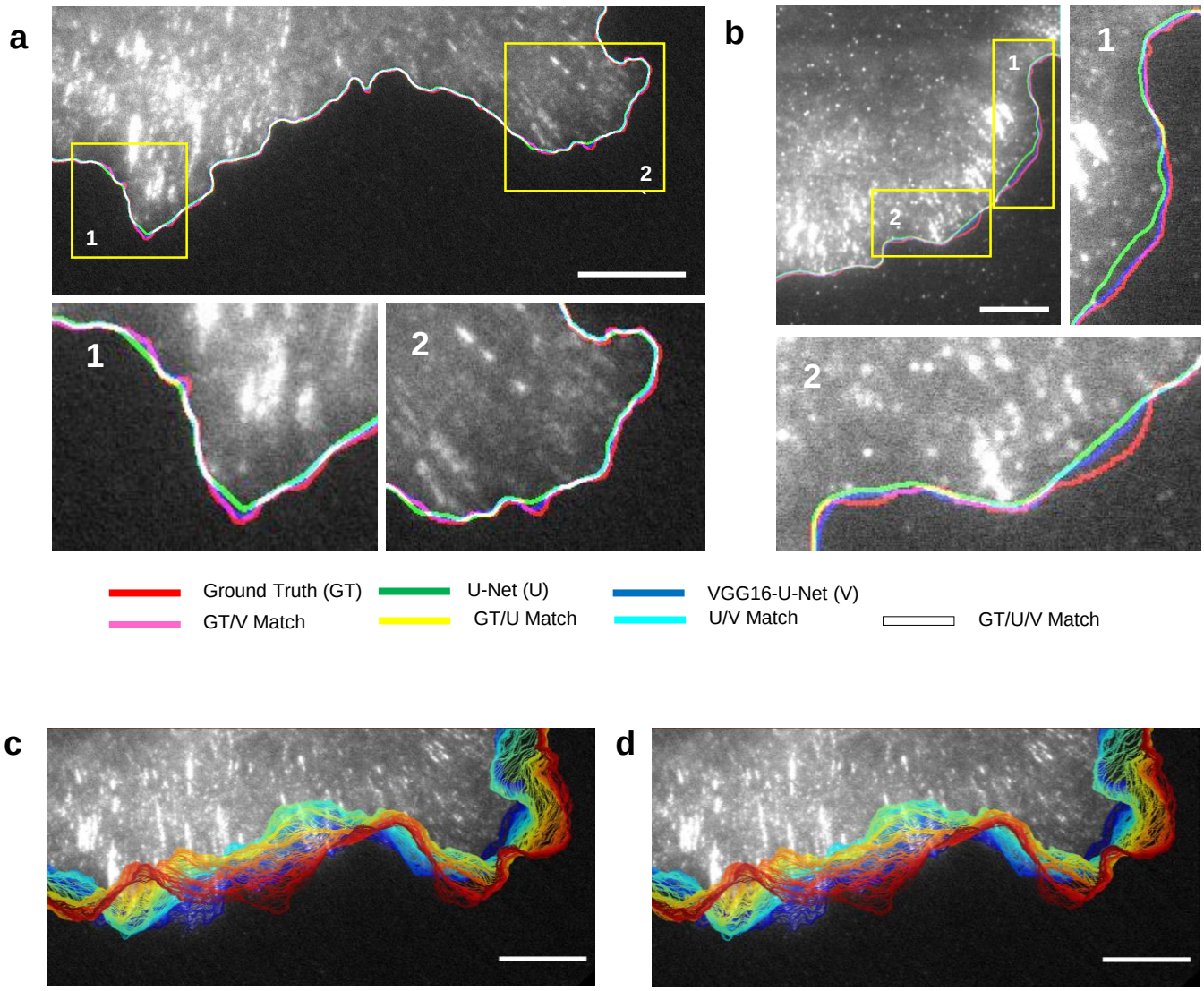


Figure 7. Visualization of segmentation results of U-Net and VGG16-U-Net for high-resolution fluorescence movies of the leading edge of a PtK1 cell expressing paxillin-HaloTag from a TIRF microscope (the number of training frames: 26). (**a**-**b**) Examples of overlaid edges of ground truth, U-Net, and VGG16-U-Net. (**c**-**d**) Overlay of edges (blue, 0s; red, 1000s time points) from U-Net (c) and VGG16-U-Net (d). Bars: 10 μm.