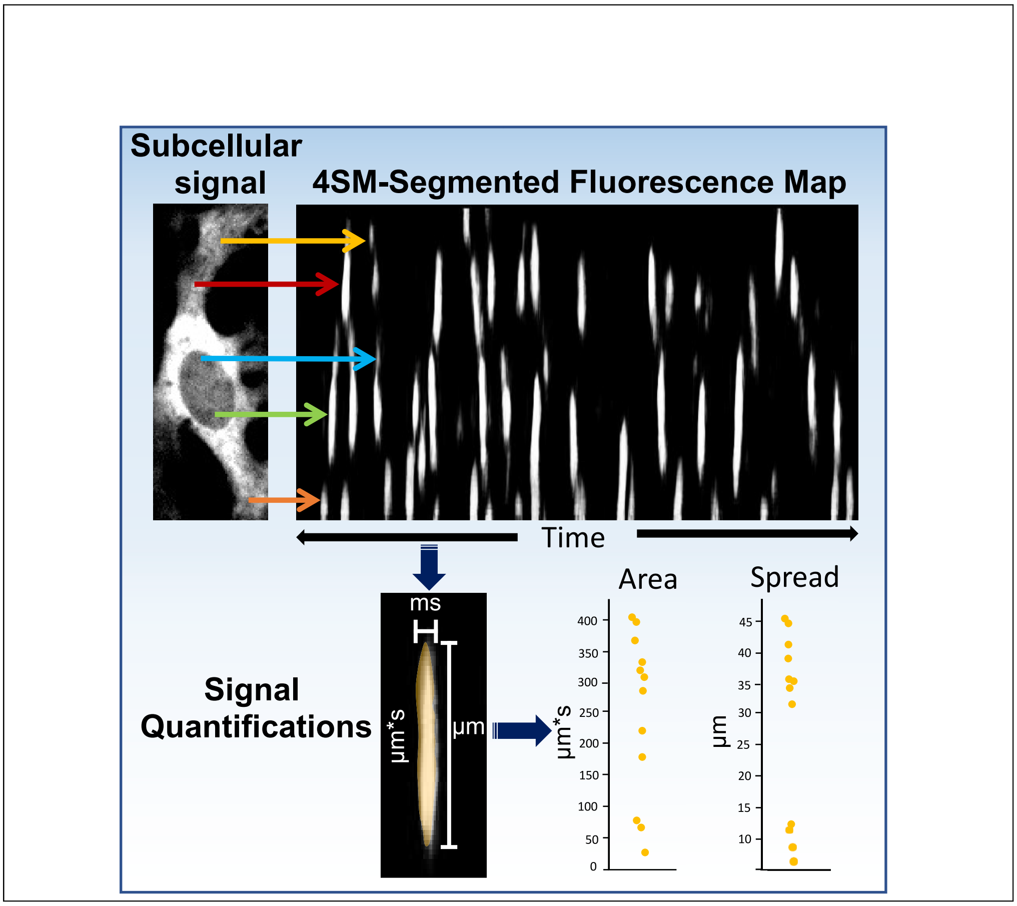
iScience

CellPress
OPEN ACCESS

Article

New open-source software for subcellular segmentation and analysis of spatiotemporal fluorescence signals using deep learning



Sharif Amit Kamran, Khondker Fariha Hossain, Hussein Moghnieh, ..., Alireza Tavakkoli, Kenton M. Sanders, Salah A. Baker

sabubaker@med.unr.edu

Highlights

4SM: an automated software solution for cellular dynamic fluorescence signal analysis

4SM relies on a novel machine-learning pipeline for fluorescence signal segmentation

4SM is fast and provides a consistent method for high-throughput analysis of datasets

4SM provides instant signal quantification and graphical representation of the results

Kamran et al., iScience *25*, 104277
May 20, 2022 © 2022 The Author(s).
https://doi.org/10.1016/j.isci.2022.104277

iScience

CellPress OPEN ACCESS

Article

New open-source software for subcellular segmentation and analysis of spatiotemporal fluorescence signals using deep learning

Sharif Amit Kamran,[1,2] Khondker Fariha Hossain,[2] Hussein Moghnieh,[3] Sarah Riar,[1] Allison Bartlett,[1] Alireza Tavakkoli,[2] Kenton M. Sanders,[1] and Salah A. Baker[1,4,*]

SUMMARY

Cellular imaging instrumentation advancements as well as readily available optogenetic and fluorescence sensors have yielded a profound need for fast, accurate, and standardized analysis. Deep-learning architectures have revolutionized the field of biomedical image analysis and have achieved state-of-the-art accuracy. Despite these advancements, deep learning architectures for the segmentation of subcellular fluorescence signals is lacking. Cellular dynamic fluorescence signals can be plotted and visualized using spatiotemporal maps (STMaps), and currently their segmentation and quantification are hindered by slow workflow speed and lack of accuracy, especially for large datasets. In this study, we provide a software tool that utilizes a deep-learning methodology to fundamentally overcome signal segmentation challenges. The software framework demonstrates highly optimized and accurate calcium signal segmentation and provides a fast analysis pipeline that can accommodate different patterns of signals across multiple cell types. The software allows seamless data accessibility, quantification, and graphical visualization and enables large dataset analysis throughput.

INTRODUCTION

In recent years there has been a surge in the use of automated-computer-aided detection for biomedical image processing and analysis (Cai et al., 2016; De Vos et al., 2016; Leigh et al., 2020; Roth et al., 2015; Teramoto et al., 2016). Automated biomedical image analysis incorporates advanced machine learning methods combined with computer vision and image processing techniques (Becker et al., 2013). Deep learning methods have shown encouraging results, outperforming experts in the field of medical imaging (Ganin and Lempitsky, 2014). Deep learning has also improved effectiveness of interpreting various modalities of data due to its computational and automated feature extraction abilities. Most biomedical images are of higher dimensions, i.e., medical resonance imaging, computed tomography, and Ca^{2+} imaging, and can be challenging and time-consuming to annotate manually. In addition, the acquiring devices from which the images are obtained may differ in their image output quality, therefore, are less optimal for deploying neural networks for such a task that requires a huge amount of consistent data for extracting features and generalizations. These limitations prompted the need for novel developments of effective systems to automate biomedical image tasks.

Generative adversarial networks (GANs) are a recent machine-learning-based approach highly utilized in different image-related applications such as image translation (Chen and Hays, 2018; Sangkloy et al., 2017), editing (Dekel et al., 2018; Zhu et al., 2016), and image style transfer (Wang et al., 2018; Xian et al., 2018). GANs are categorized into two architectures called generators and discriminators, which are pitted against each other to improve their learning. The purpose of the discriminator is to classify the input image as a real or fake image, whereas the generator tries to synthesize images that are realistic to fool the discriminator, like a typical min-max game. GANs can potentially extract and learn fine and coarse information from images by combining multiple architectures with multiscale resolutions (Brown and Lowe, 2003; Burt and Adelson, 1987). Such examples are widespread in both conditional (Denton et al., 2015; Huang et al., 2017) and unconditional GAN settings (Chen and Koltun, 2017; Zhang et al., 2017). By incorporating multiple high-resolution architectures, they can learn distinct domain-specific features with high precision and robustness. Recent advancements in deep learning architectures including

[1] Department of Physiology and Cell Biology, University of Nevada, Reno School of Medicine, Anderson Medical Building MS352, Reno, NV 89557, USA

[2] Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557, USA

[3] Department of Electrical and Computer Engineering], McGill University, Montréal, QC H3A 0E9, Canada

[4] Lead contact

*Correspondence: sabubaker@med.unr.edu

https://doi.org/10.1016/j.isci.2022.104277

pix2pixHD (Wang et al., 2018), SPADE (Park et al., 2019), and Stargan-v2 (Choi et al., 2020) made way for the synthesis of high fidelity and vivid adversarial images.

In addition, cross-domain image translation is another technique utilized that generates images from incomplete source information. It has been employed in medical image inpainting for standard statistical analysis (Dalca et al., 2018; Eilertsen et al., 2008; Van Tulder and de Bruijne, 2015), to improve standard steps of examination such as image registration (Iglesias et al., 2013; Wang et al., 2020a), to fuse information (Du et al., 2016; He et al., 2020; Wang et al., 2020b), in image segmentation (Chartsias et al., 2018; Dekel et al., 2018; Li et al., 2020; Roy et al., 2011), image construction (Commowick et al., 2009; Cordier et al., 2016), and in disease diagnosis (Li et al., 2014; Zhou et al., 2020). Each of these techniques convert images of one modality to another interchangeably. Some examples include magnetic resonance imaging (MRI), optical coherence tomography (OCT), spectral-domain OCT, positron emission tomography (PET), and ultrasound imaging (Dalca et al., 2018; Dekel et al., 2018; Eilertsen et al., 2008; Iglesias et al., 2013; Li et al., 2020; Van Tulder and de Bruijne, 2015; Wang et al., 2020a, 2020b).

The ideal system is one that would have a dynamic end-to-end architecture in place that could both synthesize biomedical images from one modality to another and extract from and learn representations of the manifold features. In this paper we propose a new subcellular segmenting and analyzing tool for dynamic fluorescent signal maps that utilizes core generative architecture for segmenting dynamic fluorescent events (i.e., Ca^{2+} transient signals).

Dynamic fluorescence imaging is a widely used tool in medical research and provides valuable information on cellular function and regulation. Dynamic fluorescent signals can be monitored at a cellular level to visualize changes in ions, pH, protein trafficking, or voltage in many pathologies. These subcellular signals are complex, for example, voltage and Ca^{2+} signaling patterns differ depending on cell type, cellular compartmentalization, and target tissue (Roome and Kuhn, 2018). Ca^{2+} signaling can range from intercellular and intracellular waves that spread for long distances, over several microns or more, to local Ca^{2+} release events that spread on the scale of nanometer to micrometer range (Baker et al., 2013; Berridge and Dupont, 1994; Cheng et al., 1996; Drumm et al., 2015; Hennig et al., 2010; Straub et al., 2000). Monitoring Ca^{2+} signaling can provide an in-depth approach to understanding the complexities of essential cellular activities. Extracting important information from cellular Ca^{2+} signals such as Ca^{2+} spread, duration, and initiation sites is almost entirely dependent on segmentation of Ca^{2+} signals during analysis. Ca^{2+} STMaps are widely used for the analysis and quantification of voltage and Ca^{2+} dynamics parameters that retain Ca^{2+} event information as a function of space occupied over time (Cheng et al., 1996; Colman et al., 2017; Hennig et al., 1999; Lee et al., 2009; Lentle and Hulls, 2018; Roome and Kuhn, 2018; Waadt et al., 2017). As a result, STMaps can provide a platform to effectively extract quantifiable cellular Ca^{2+} data (Baker et al., 2021a, 2021b; Drumm et al., 2014; Fedigan et al., 2017; Sancho et al., 2017; Sergeant et al., 2006). Extracting dynamic fluorescent data can be a challenging task, as the information is most often manually defined using single-line pixel measurement or region of interest (ROI). Because of this, discrepancies between users and individual user error are high, which can translate to inaccuracies during STMap event quantification. The current user-dependent process is highly variable, time-consuming, and labor-intensive; therefore, we propose a new software for efficient end-to-end architecture to precisely segment and quantify dynamic fluorescent cellular signals in a fast automated fashion. The tool will allow researchers to swiftly analyze large volume datasets with high accuracy.

RESULTS

Subcellular signal segmentation using generative adversarial networks architecture

We used a state-of-the-art generative adversarial networks architecture to establish the proposed Spatio-temporal Subcellular Signal Segmentation Model “4SM”. The model incorporates multiple novel components to provide a powerful and consistent subcellular signal segmentation tool, as discussed further below.

Subcellular signal datasets

Datasets from subcellular Ca^{2+} signals imaged from gastrointestinal pacemaker-type cells called interstitial cells of Cajal (ICC) from the colon, stomach, and small intestine of Kit-Cre-GCaMP6f mice were used to train the 4SM model (Figure 1A). Ca^{2+} signals provide excellent variable patterns ranging from stochastic to rhythmic subcellular signals that can provide the basis of segmenting subcellar fluorescent signals in our

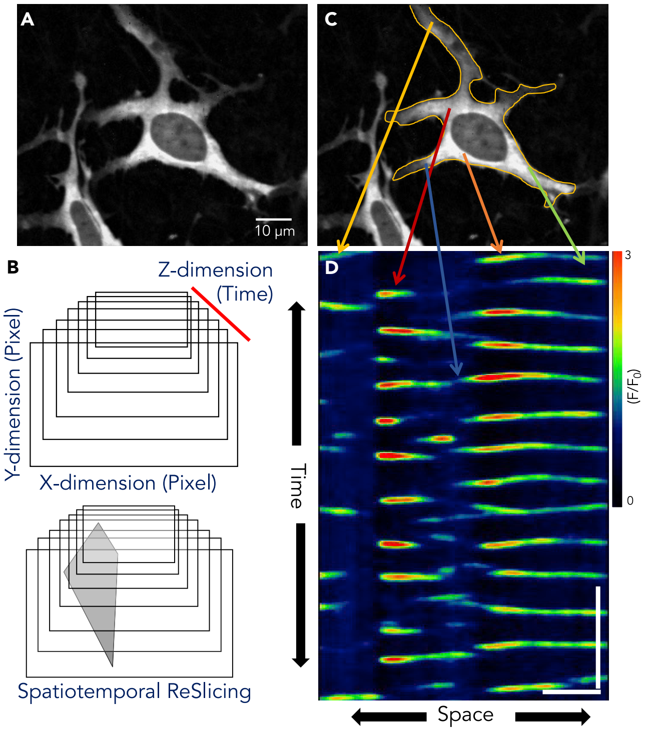


Figure 1. Cellular Ca^{2+} spatiotemporal maps (STMaps)

(A) Representative image of subserosal ICC (ICC-SS) from the proximal colon of a Kit-Cre-GCaMP6f mouse *in situ*. The calcium sensor GCaMP6f is expressed in ICC cells. Scale bar in A is 10 μm.

(B) Image sequences of fluorescent signal (GCaMp6f) are acquired and presented as stacks or a movie. The stacks are composed of three dimensions x and y for each frame and z dimension, where z is time. Spatiotemporal two-dimensional map is constructed because of spatiotemporal reslicing of an X-Y-Z section in a movie or image stack to generate fluorescence STMaps.

(C) A single ICC-SS within the FOV (outlined in orange) and defines the cellular ROI for Ca^{2+} signal. Colored arrows from the cell correspond to the locations of each Ca^{2+} sites within the cell and plotted in the STMap in D.

(D) Representative two-dimensional color-coded intensity STMap of Ca^{2+} activity in ICC-SS. Each horizontal unit (cell space) in the STMaps represent a pixel or average of pixels in the original image. The coded pixels from each frame of the movie were used to construct a single row, and sequential rows were placed under each other to produce an STMap. Fluorescence intensity is plotted as (F/F$_0$). Scale bars in D are x = 10 μm and y = 2.5s.

model. Dynamic Ca^{2+} signals can then be plotted and visualized as STMaps. The 2D map is created from spatiotemporal reslicing of an X-Y-Z section in a movie or image stack (Figure 1B), and STMaps can be used for fluorescent signal analysis in a variety of cell types including neurons, cardiac myocytes, and smooth muscle cells. In our dataset, Ca^{2+} signals in ICC were recorded using spinning-disk confocal microscopy (Figures 1A and 1C). One cell type within the colon, subserosal ICC (ICC-SS), exhibited variable and complex patterns of Ca^{2+} signals (Figure 1). STMaps are a valuable tool because they allow for a more complete representation of individual cell Ca^{2+} signals through plotting Ca^{2+} events in cell space (x-axis) and temporally (y-axis) (Figure 1D). Therefore, Ca^{2+} signal features and quantification from STMaps can effectively describe the cellular Ca^{2+} dynamics and behaviors in many cell types. Accurate and successful analysis of Ca^{2+} imaging data and other fluorescent dynamic signals from STMaps depends heavily on defining and segmenting diverse signal events. Therefore, we created a novel deep-learning method that employs generative adversarial network architectures to segment signals with high accuracy.

Proposed generators of 4SM

4SM is composed of two generators, each consisting of various block components that effectively carries out the fine-grained segmentation process. Combining two generators for fine and coarse feature distinction is highly effective and has shown notable visual results for image translation tasks. We used two generators in our proposed method, G$_{\text{fine}}$ and G$_{\text{coarse}}$ (Figure 2). G$_{\text{fine}}$ learns local information of the event's

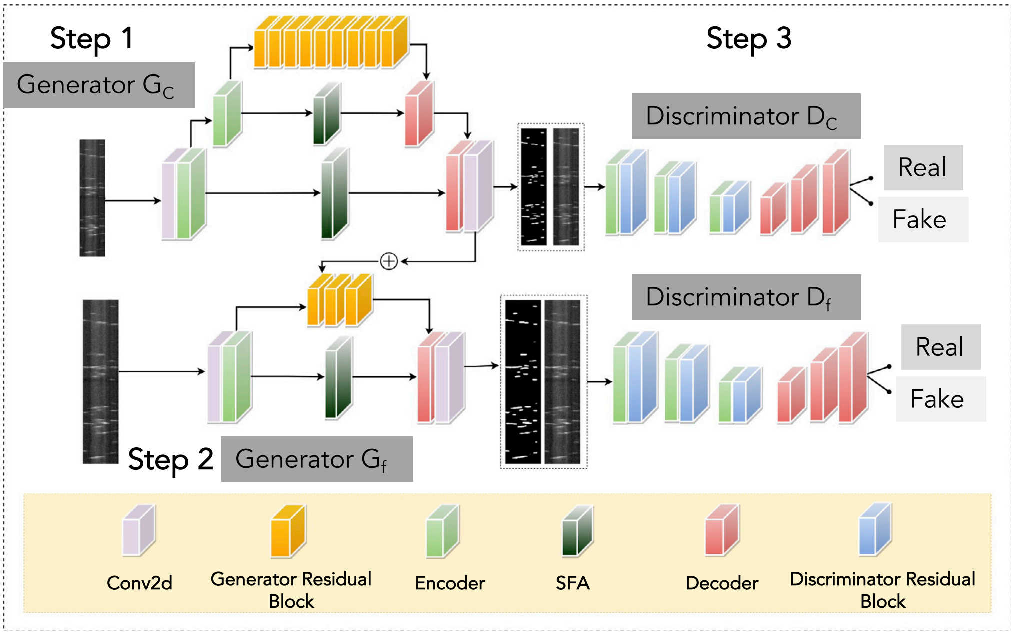


Figure 2. Overall architecture design for 4SM

Proposed generative adversarial network, 4SM, consists of two generators, G_{coarse} (G_c) and G_{fine} (G_f), and two discriminators, D_{coarse} (D_c) and D_{fine} (D_f). Step 1: the coarse generator takes smaller Ca^{2+} STMaps as input and outputs coarsely segmented STMaps. Step 2: the fine generator takes larger Ca^{2+} STMaps as input and outputs finely segmented STMaps. Step3: fine and coarse discriminators take both original Ca^{2+} STMaps and segmented Ca^{2+} STMaps as input and outputs a probability map, which dictates if the pairs are either real or fake. The architecture also contains spatial-feature-aggregation (SFA) blocks in both the generators for extracting manifold features and synthesizing realistic images and two-dimensional convolution (Conv2d). The discriminators use auto-encoders for generated pixel-wise output to dictate if each pixel is real or fake.

spatial spread and duration, whereas G_{coarse} learns the boundary between multiple occasions in the generated Ca^{2+} STMaps. These generators are made of multiple residually connected blocks, downsampling and upsampling layers, and a skip connection of features between the fine and coarse generators. There is a manifold of inherent features that can be extracted from the STMaps, as shown in (Figures 1 and 6). To extract those features and to retain the visual information, we incorporated a spatial feature aggregation block as illustrated in Figures 2 and 3. G_{fine} has an input resolution of 64 × 64 × 1 and outputs images with the same resolution. Likewise, G_{coarse} takes an image with half the resolution (32 × 32 × 1) and produces an image with the same resolution. Furthermore, the G_{coarse} outputs a feature vector size 32 × 32 × 64 that is combined with one of the intermediate layers of G_{fine} using a skip connection module between the generators. In the next sections, we discuss each of these blocks in detail.

Upsampling and downsampling blocks

Using convolution-based upsampling and downsampling may result in information loss and an inability to retain specific spatial information. However, we've created two new optimized blocks for these operations (Figures 3A and 3B). These two new upsampling and downsampling blocks can retain intrinsic characteristics and tackle information loss due to spatial feature compression. Both generators and discriminators of the 4SM model incorporates the downsampling block and only the generators use the upsampling block to generate the feature maps and output. The downsampling block comprises a convolution layer followed by a Batch-Normalization layer (Ioffe and Szegedy, 2015) and Leaky-ReLU activation layer (Figure 3A). Conversely, the upsampling block contains a transposed convolution layer, followed by Batch-Normalization (Ioffe and Szegedy, 2015) and Leaky-ReLU activation layer as illustrated in (Figure 3B). The downsampling block is used twice in G_{coarse} after consecutive residual blocks, and the upsampling block is used twice to get the original spatial dimension's feature output. For G_{fine}, the downsampling block is used once, and after multiple repetitions of residual blocks, a single upsampling block is used to get the original spatial dimension output. For both convolution and transposed convolution functions, a kernel size, K = 3, and stride, S = 2, was used.

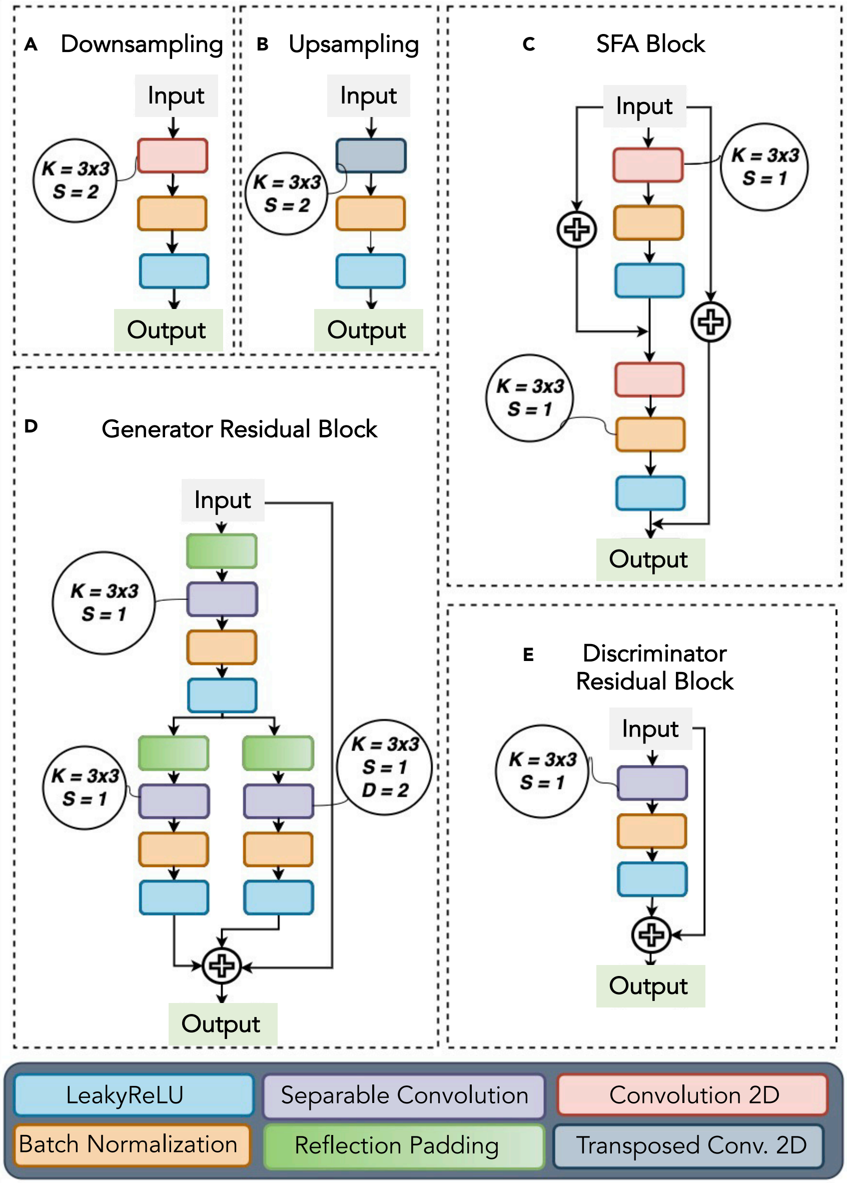


Figure 3. Building blocks for generators and discriminators

The individual blocks of 4SM architecture consist of (A) downsampling block, (B) upsampling block, (C) spatial feature aggregation block, (D) residual generator block, and (E) residual discriminator block, where K stands for kernel size, S is for stride, and D is for dilation rate. The "+" indicates element-wise summation of features in the depth axis.

Residual block

The basic configuration of our 4SM model comprises a residual unit with two successive separable convolution layers and a skip connection that adds the input feature tensor with the output. Regular convolution layers are computationally inefficient compared with separable convolution (Chollet, 2017). The main difference between regular convolution and separable convolution is that the latter incorporates a depth-wise convolution followed by a point-wise convolution. By utilizing separable convolution, the depth and spatial information is retained from the data. Some recent findings reported that combining separable convolutional layers with dilation allows for a more robust feature acquisition (Kamran et al., 2020). By incorporating this idea, we designed the proposed residual block to retain both depth and spatial information by using separable convolution followed by Batch-Normalization and Leaky-ReLU as a postactivation mechanism to decrease the number of computational parameters and ensure effective memory utilization (Figure 3D).

Residual block reflection padding was also incorporated before each separable convolution operation to pad plausible data values by reusing values present along the borders of the input. After numerous experimentations, we found that a wider receptive field optimally captured neighboring information. To adjust for this, the residual block was further modified to contain two branches of separable convolution layers with distinct dilation rates, as illustrated in (Figure 3D); one branch with a dilation rate of D = 1 and the other with a dilation rate D = 2. A kernel size, K = 3, and stride, S = 1, was used for all separable convolution layers where each separable convolution has a pre-reflection padding layer, post-Batch-Normalization and Leaky-ReLU activation layer. Finally, the skip connection from the input and the two branches' output was combined collectively to generate the final output. In addition, a similar technique was incorporated to create a distinct residual block for the discriminators, as shown in (Figure 3E). Conversely, the residual block for the discriminator is made of a separable convolution layer, followed by Batch-Normalization and Leaky-ReLU activation functions where the separable convolution has a kernel size of K = 3 and stride, S = 1.

Spatial feature aggregation block

The proposed spatial feature aggregation (SFA) block (Figure 3C) was implemented to accommodate for the potential loss of spatial information from the manifold feature space due to consecutive upsampling and downsampling (Chen et al., 2018; Zhang et al., 2019). To retain intrinsic features, the skip connections combined with the SFA block were used to fuse feature information from the bottom layers of the network with the top layers. The block comprises two successive residual units with convolution, Batch-Normalization and Leaky-ReLU layers. Batch-Normalization is utilized for faster and more stable training by normalizing the inputs with re-centering and re-scaling (Ioffe and Szegedy, 2015). Leaky-ReLU is used for solving the vanishing gradient problem (Maas et al., 2013). Here, the kernel size is K = 3 and stride, S = 1. There are two skip connections: (1) coming from the input and added to the first residual unit's output and (2) coming from the input and element-wise summed with the last residual unit's output. G_{coarse} consists of two SFA blocks that come out of two encoders and are successively added with the two decoders. Conversely, G_{fine} has only one SFA block between the encoder and decoder.

Autoencoders as discriminators

In the proposed subcellular signal segmenting model, we used two discriminators, one for G_{coarse} and one for $G_{fine,}$ termed D_{coarse} and D_{fine}, respectively. The decoders were incorporated to convert the discriminators into a single auto-encoder, unlike the patchGAN discriminators seen in other architectures (Choi et al., 2020; Park et al., 2019; Wang et al., 2018). The auto-encoder dictates if each pixel, rather than patches, of the generated image is real or fake. This in turn helps with the pixel-level segmentation task, as more discriminative features are retained throughout the architecture and results in a highly accurate segmentation of signals in STMaps. D_{fine} interprets STMaps with an input size of 64 × 64 × 1 and D_{coarse} interprets STMaps with an input size of 32 × 32 × 1 and outputs feature-maps of 64 × 64 × 1 and 32 × 32 × 1 successively. Each point in this feature vector represents values between −1 and +1. Here, −1 represents a real value and +1 represents a synthesized or adversarial pixel.

Adversarial objective function

To promote adversarial training, a multihinge loss for GANs (Lim and Ye, 2017; Zhang et al., 2019) was used in our architecture as illustrated in Equations 1 and 2. Conclusively, all the Ca^{2+} STMaps and their corresponding masks were normalized to [−1,1] to widen the gap between the real and synthesized segmentation map's pixel intensities. In Equation 3, we added them and used λ_{adv} as a weight multiplier with the $L_{adv}(G)$.

$$L_{adv}(D) = -E_{x,y}[min(0, -1 + D(x,y))] - E_x[min(0, -1 - D(x,G(x)))] \quad \text{(Equation 1)}$$

$$L_{adv}(G) = -E_{x,y}[(D(G(x),y))] \quad \text{(Equation 2)}$$

$$L_{adv}(G,D) = L_{adv}(D) + \lambda_{adv}(L_{adv}(G)) \quad \text{(Equation 3)}$$

In Equation 1, the discriminators were first trained on real STMaps, *x*, and real segmentation maps, *y*, which is signified by *D (x,y)* and then retrained on the real STMaps, *x*, and the synthesized segmentation map, *G(x)*, which is signified by *D (x, G(x))*. The $E_{x,y}$ stands for expected value of multiple samples, given the input *x,y and min* is for taking the minimum value out of the two values inside the parenthesis. First, the discriminators D_{fine} and D_{coarse} went through batch-wise training for several iterations on randomly sampled data. Following that, the G_{coarse} was trained while the weights of D_{fine} and D_{coarse} were kept isolated. In the same

Table 1. Five-fold cross-validation of 4SM segmentation masks

Fold	Mean-IoU	SSIM	Dice-coeff
1	92.98	89.55	96.31
2	92.96	89.60	96.30
3 (Best)	93.06	89.52	96.36
4	92.92	89.58	96.28
5	92.81	89.44	96.17

manner, the G_{fine} was trained on a batch of random STMaps, whereas the weights of both the D_{fine} and D_{coarse} were kept isolated. The generators incorporated the reconstruction loss as shown in Equation 2. By utilizing these losses, the generated segmentation map depicted a more realistic representation of actual Ca^{2+} events. We also incorporated a feature-matching loss component (Wang et al., 2018) with the D_{fine} and D_{coarse}, given in Equation 5.

$$L_{rec}(G) = E_{x,y}\|(G(x) - y)^2\| \qquad \text{(Equation 4)}$$

$$L_{fm}(G, D_n) = E_{x,y}\sum_{i=1}^{k}\frac{1}{N}\|D_n^i(x,y) - D_n^i(x, G(x))\| \qquad \text{(Equation 5)}$$

Where, L_{rec} is the reconstruction loss for a real STmap, y, given a generated segmentation map, $G(x)$, and $\|\cdot\|$ stands for absolute value. The loss for both G_{fine} and G_{coarse} was utilized so to allow our model to generate high-quality segmentation maps of two distinct scales. This technique has been previously utilized by combining GAN cost function with a mean squared error (MSE) loss, a commonly used loss function for regression (Pathak et al., 2016). Finally, Equation 6 was estimated by first taking the features from the intermediate layers of the discriminators by inserting the real and then the synthesized segmentation map sequentially. Here, N stands for the number of feature layers extracted from the discriminators, $D_n(x,y)$ is the intermediate feature from discriminator with real image as input, and $D_n(x, G(x))$ is the intermediate feature from discriminator with synthesized image as input. By joining Equations 3, 4, and 5 the final objective function for 4SM was formulated as shown in Equation 6.

$$\min_{G_f,G_c}((L_{adv}(G_f, G_c, D_f, D_c, \lambda_{adv})) + \lambda_{rec}[L_{rec}(G_f, G_c)] + \lambda_{fm}[L_{fm}(G_f, G_c, D_f, D_c)]) \qquad \text{(Equation 6)}$$

Here, λ_{adv}, λ_{rec}, and λ_{fm} signify loss weighting that were multiplied with their corresponding losses. The loss weighting dictates which networks to focus on while training. For the proposed architecture, more weight is given to the λ_{adv} and λ_{rec}, and thus the bigger λ values are selected.

Segmenting model experimentation

Dataset validation

We trained our 4SM model on STMaps generated from 64 Ca^{2+} imaging movies with varied resolution, taking 64 × 64 overlapping crops with stride 8. Furthermore, we followed the same procedure to test on STMaps generated from 17 additional videos. The images were collected in gray-scale format, and the segmentation was performed in a binary format. We used the leave-one-out of cross-validation (LOO-XVE) for training the architecture and separated the dataset into 5 folds. This technique involved training on 4 folds and validating on the remaining single fold. We tested all 5 versions of the model on the test dataset and selected the most accurate as the primary model (Table 1).

Stride and inference speed

STMaps can vary in image size, so we divided STMap images into smaller images by cropping with a fixed window size of 64 × 64 resolution for robust segmentation of Ca^{2+} events. The cropping window goes from left-to-right and then top-to-bottom and can overlap on top of each other. This overlapping factor is called stride or step-size. For example, given image size = 128 × 128, if we choose stride = 64, then the number of cropped images will be 4. If the stride = 8, the cropped images will be 81. The equation to find the number of crops per image can be calculated as follows:

$$Num\_of\_crops = \left[\frac{I_{Height} - C}{S} + 1\right] \times \left[\frac{I_{width} - C}{S} + 1\right] \qquad \text{(Equation 7)}$$

In Equation 7, I = image height or width, C = crop size, S = stride, and O = number of crops for height or width. The equation shows that a larger stride will create a small number of crops from each test image, and a smaller stride will yield a larger number of cropped images. Contrarily, larger strides will speed up the model's prediction but will result in poor visual quality. At the same time, smaller strides will slow down the model's prediction and yield a high-quality visual result. One of the features of our model is that the user can control this stride parameter while testing their images.

Hyper-parameter selection

For adversarial training, hinge loss was used for our segmentation model (Lim and Ye, 2017; Zhang et al., 2019). We selected $\lambda_{adv} = 10$, $\lambda_{rec} = 10$, and $\lambda_{fm} = 1$ (Equation 6). For optimizer, Adam (Kingma and Ba, 2014) was used, with a learning rate of $\alpha = 0.0002$, $\beta_1 = 0.0002$, and $\beta_2 = 0.999$. For training our model, a batch of $b = 64\ was$ trained for 100 epochs, which took approximately 12 h to train on NVIDIA P100 GPU.

Training

To train our segmentation model, multiple steps were generated: (1) all the hyper-parameters were initialized, (2) a sample batch of the real STMaps and segmentation map, x, and y were created; (3) the real STMaps were trained and fake segmentations were paired with D_{fine} and D_{coarse}; (4) the G_{fine} and G_{coarse} were used to synthesize fake segmentation maps and the real STMaps and fake segmentation-map, x, G(x) were used to train discriminators D_{fine} and D_{coarse}; (5) the adversarial loss was calculated, $L_{adv}(D, G)$, and weights were updated, while keeping the weights of the discriminators isolated; and (6) the generators were trained. We calculated the $L_{rec}(G)$ reconstruction loss and updated both generator's weights. Subsequently, both discriminators' weights were reintegrated, the feature matching loss $L_{fm}(D)$ was calculated and the discriminator weights were then updated. In the final stage, both discriminator's weights were isolated and together, the discriminators and generators were adjusted. Lastly, total loss was calculated by adding and multiplying their relative weights in the model, and weights were recorded.

Evaluation and result

We refer to the newly proposed Spatiotemporal Subcellular Signal Segmenting Model as 4SM, and part of its novelty is due to its ability to pick up and discriminate subtle differences in Ca^{2+} events in a variety of cell types. Original STMaps for both rhythmic and stochastic Ca^{2+} style events were produced from two different subsets of ICC's as shown in (Figure 4). The original STMaps (Figures 4 *Ai and Bi*) were segmented by our model (Figures 4 *Aii* & *Bii*), and the overlays are visualized in Figures 4 *Aiii* & *Biii*. To evaluate the segmentation efficiency of our architecture, we color-coded the original STMaps in red and the 4SM segmented maps in green to allow better visualization of Ca^{2+} events and overlap in yellow (Figures 4C and 4D). The area overlap percentages of rhythmic signals were 81.5% ± 2.1% (n = 5) and 86% ± 3.9% (n = 5) for stochastic Ca^{2+} signals. We also calculated the Pearson's Correlation Coefficient (PCC), which shows the close correlation of overlap between the original image and the 4SM segmented image. PCC values were 0.94 for rhythmic signals and 0.86 for stochastic signals (Figures 4E and 4F; n = 5).

The proposed 4SM model was compared with current state-of-the-art segmentation models for Ca^{2+} STMap segmentation based on Weka segmentation models (Leigh et al., 2020) and a deep-learning-based auto-encoder model called U-Net (Ronneberger et al., 2015). The Weka model utilized a fast-random-forest for segmenting Ca^{2+} events with hand-selected features to generate binary segmentation of the Ca^{2+} transient events, whereas U-Net utilizes an encoder and decoder with convolution filters to learn spatial information from images and successively down-samples, up-samples, and extract features of different depths. U-Net has been extensively used in various biomedical imaging modalities (Alom et al., 2018, 2019; Zhuang, 2018). A side-by-side comparison of the new proposed 4SM, Weka, and U-Net predictions are given in (Figures 5A–5E). The output of the 4SM architecture produced more accurate segmentation maps that were 94% similar to the ground-truth (GT). Contrarily, Weka produced broken patches of Ca^{2+} events, and the boundaries of the events were not uniformly spread and therefore less accurate. U-Net cannot produce well-defined boundaries for the Ca^{2+} events and misclassifies background noise as events. This, in turn, makes 4SM the more optimal model out of the two. In addition, the spatial spread and duration of the events can be predicted accurately based on the predicted output using the connected component algorithm, as illustrated in (Figure 6).

We used three metrics for evaluating the segmented Ca^{2+} STMaps: (1) mean-intersection over-union (mean-IoU), (2) structural similarity index measure (SSIM), and (3) dice-coefficient. Mean-IoU is a metric

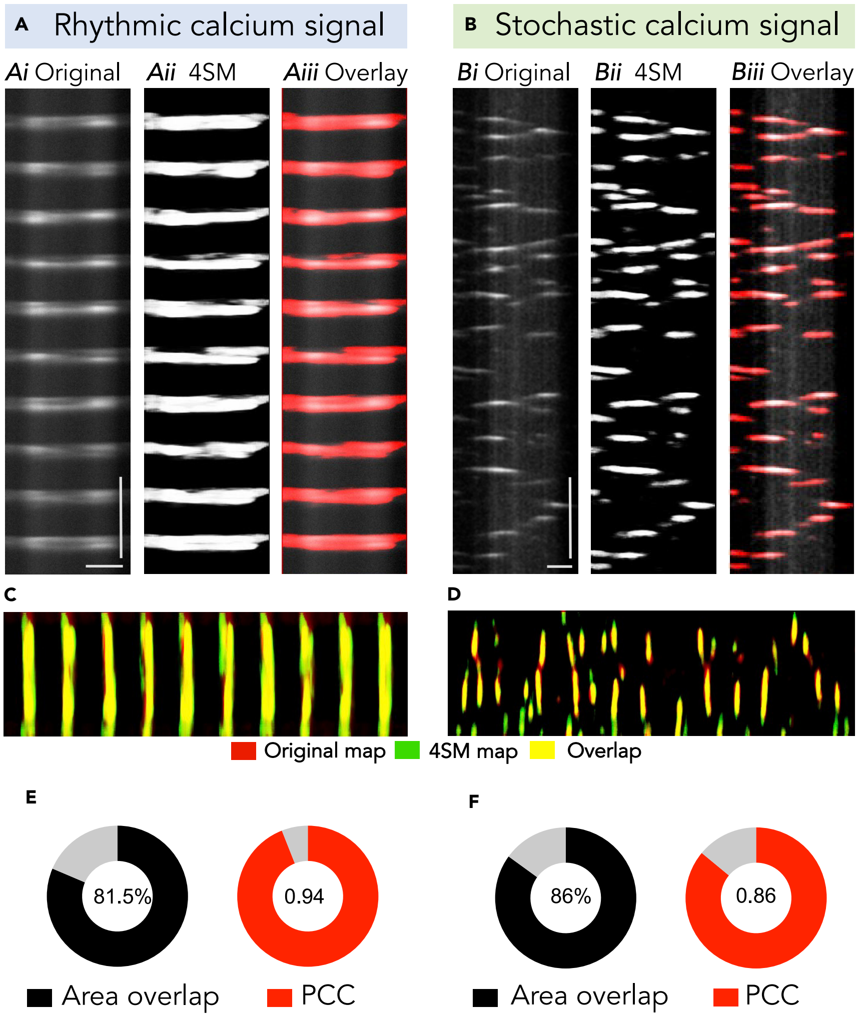


Figure 4. Predicted 4SM segmentation for Ca^{2+} STMaps

(A) Representative STMap of Ca^{2+} signals in colonic ICC-MY. (*Ai*) Original Ca^{2+} STMap; (*Aii*) predicted segmented map generated by 4SM. White pixels represent the Ca^{2+} events and black pixels are the background; (*Aiii*) semitransparent mask (red) from *Aii* overlayed on original STMap in *Ai*, which shows the accuracy and extent of the predicted mask. Scale bars in Ai are x = 10 μm and y = 5 s.

(B) Representative STMap of Ca^{2+} signals in small intestine ICC-DMP. *(Bi)* Original Ca^{2+} STMap; (*Bii*) 4SM segmented map, (*Biii*) semitransparent mask (red) from *Bii* overlayed on STMap *Bi*. Scale bars in Bi are x = 10 μm and y = 5 s.

(C) Combined map of two merged STMaps from rhythmic ICC-MY (red) and 4SM segmented map (green), and the overlay is representative of the complete pixel to pixel overlap of both maps (yellow).

(D) Two overlayed STMaps from the stochastic Ca^{2+} signals from ICC-DMP; both original (red) and 4SM segmented map (green).

(E and F) Summary data of percentage of Ca^{2+} signals area overlaps between the intracellular Ca^{2+} transients in the raw STMaps and 4SM masks, where E is for rhythmic and F is for stochastic events. Pearson's correlation coefficient (PCC) showing the linear relationship of overlap between the original STMaps and the 4SM maps (E & F; n = 5).

that measures the percentage overlap between the target mask and the prediction mask by dividing the number of pixels that overlap by the total number of pixels. Dice coefficient is calculated as 2*TP/(2 TP + FP + FN). Here, TP, FP, and FN denote true positive, false positive, and false negative, respectively. Mean-IoU and dice-coefficient are the current gold standards for measuring segmentation results for many semantic segmentation challenges such as Pascal-VOC2012 (Everingham et al., 2015), MS-COCO (Lin et al., 2014), and Cityscapes (Cordts et al., 2016). Contrarily, SSIM is a standard metric for evaluating GANs for image-to-image translation tasks by measuring image quality degradation.

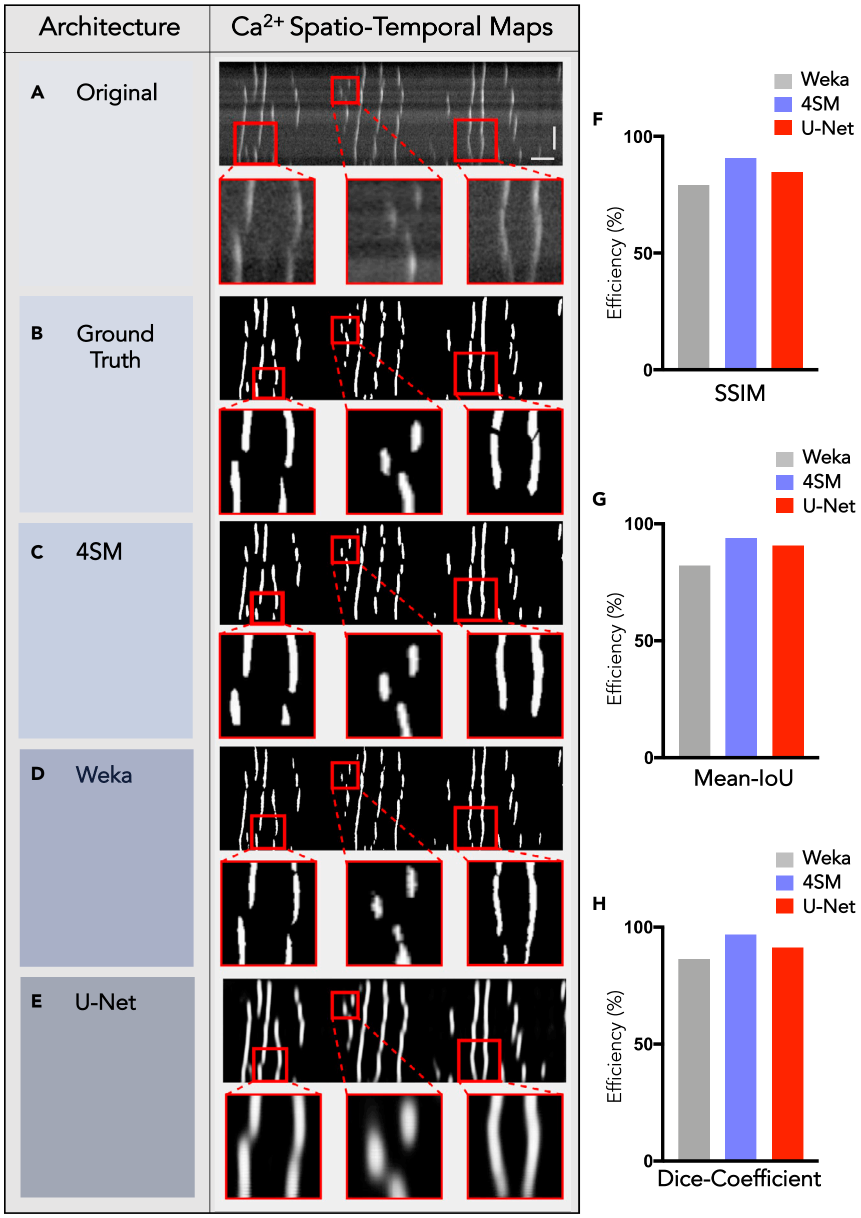


Figure 5. Comparative predicted segmentation for 4SM

(A) Representative raw STMap of Ca^{2+} signals in colonic ICC-IM.

(B) Ground-truth segmentation mask annotated by an expert.

(C) Predicted segmentation mask generated by 4SM, (D) predicted segmentation mask generated by Weka, and

(E) predicted segmentation mask generated by U-Net. The red rectangle boxes on STMaps are plotted every second row from the top and highlight the expanded scale of STMap Ca^{2+} events.

(F–H) STMap segmentation efficiency, measured as percentage efficiency, using SSIM (F), mean-IoU (G), and dice-coefficient (H) between Weka, 4SM, and U-Net architectures. Scale bars in A are x = 2 s and y = 10 µm.

More simply, SSIM predicts the perceptual difference and similarity between two similar images. We provide the 5-fold cross-validated results for our model in terms of mean-IOU, SSIM, and dice-coefficient as shown in (Table 1). The best performing model is Fold 3. For testing, we compared between Weka trained model, 4SM model (Fold −3), and U-Net. 4SM outperforms both U-Net and Weka segmentation, in terms of mean-IoU, SSIM, and dice-coefficient, the three main metrics for this task as shown in Figures 5F–5H and Table 2.

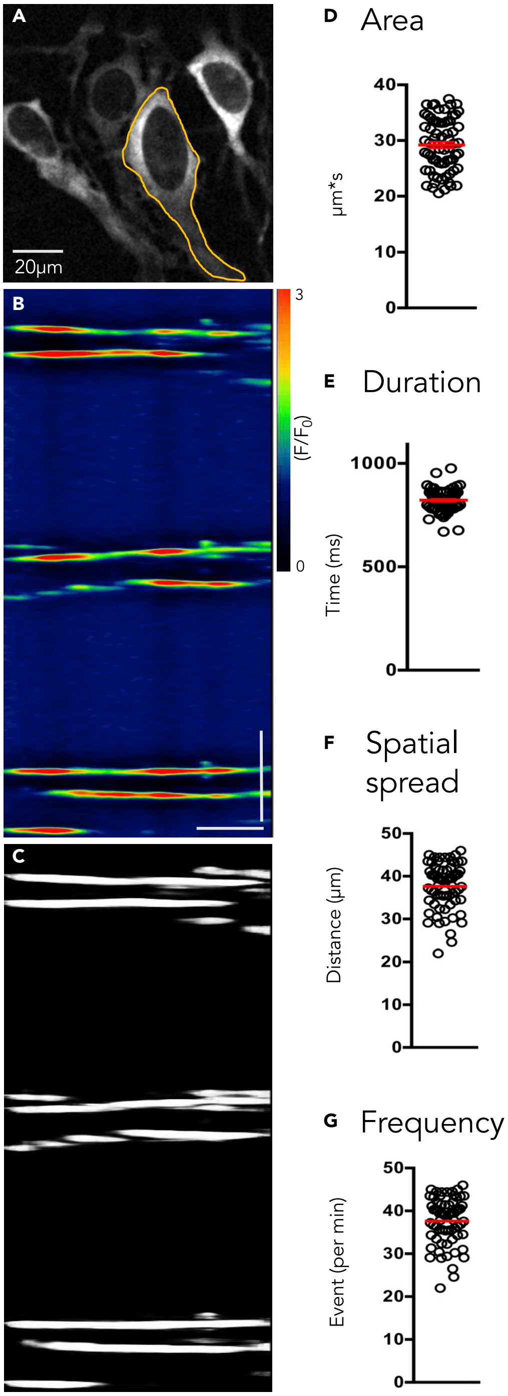


Figure 6. Quantification of segmented Ca^{2+} events

(A) Representative image of ICC-MY cells from the gastric antrum of Kit-Cre-GCaMP6f mouse; the bar in A represents 20 μm.

(B) Corresponding color-coded STMap with fluorescent intensity indicated, plotted as (F/F0). Scale bars in B are x = 10 μm and y = 5 s.

(C) Representative segmented STMap using 4SM. The quantifications of Ca^{2+} events parameters are plotted in (D–G). (D) Ca^{2+} event area, (E) Ca^{2+} event duration, (F) Ca^{2+} event spatial spread, and (G) Ca^{2+} event frequency. Data were used from various 4SM STMaps from all ICC; n = 17. The red lines in D–G represent the average mean values.

Table 2. Comparison between 4SM, Weka, and U-Net segmentation models

Method	Mean-IoU	SSIM	Dice-coeff
Weka	82.22	79.16	86.24
4SM	93.85	90.63	96.79
U-Net	90.62	84.83	91.28

Quantification of signal event parameters

4SM incorporates effective quantification of subcellular event parameters. We achieved this by applying a connected component algorithm (He et al., 2009) on the 4SM segmented STMaps. The connected component algorithm, imported from the Open-CV library (Bradski, 2000), finds the frequency of Ca^{2+} events and defines the rectangular bounding box surrounding each event to define its duration, spatial spread, and area parameters in pixels (Figures 6A–6G). We also calculated the interval duration between Ca^{2+} events by subtracting the lower-most pixel of the first event from the upper-most pixel of the second event and applied the process for the remaining events. Moreover, the tool integrates an image calibration option to multiply the pixels in terms of scales (millimeter, centimeter, etc.); this allows the user to easily quantify the pixel values to its native scale size. Implementing the quantification process of several Ca^{2+} parameters within 4SM allows effective data interpretation and enhances the analysis pipeline speed.

Implementation of 4SM into a web-based tool

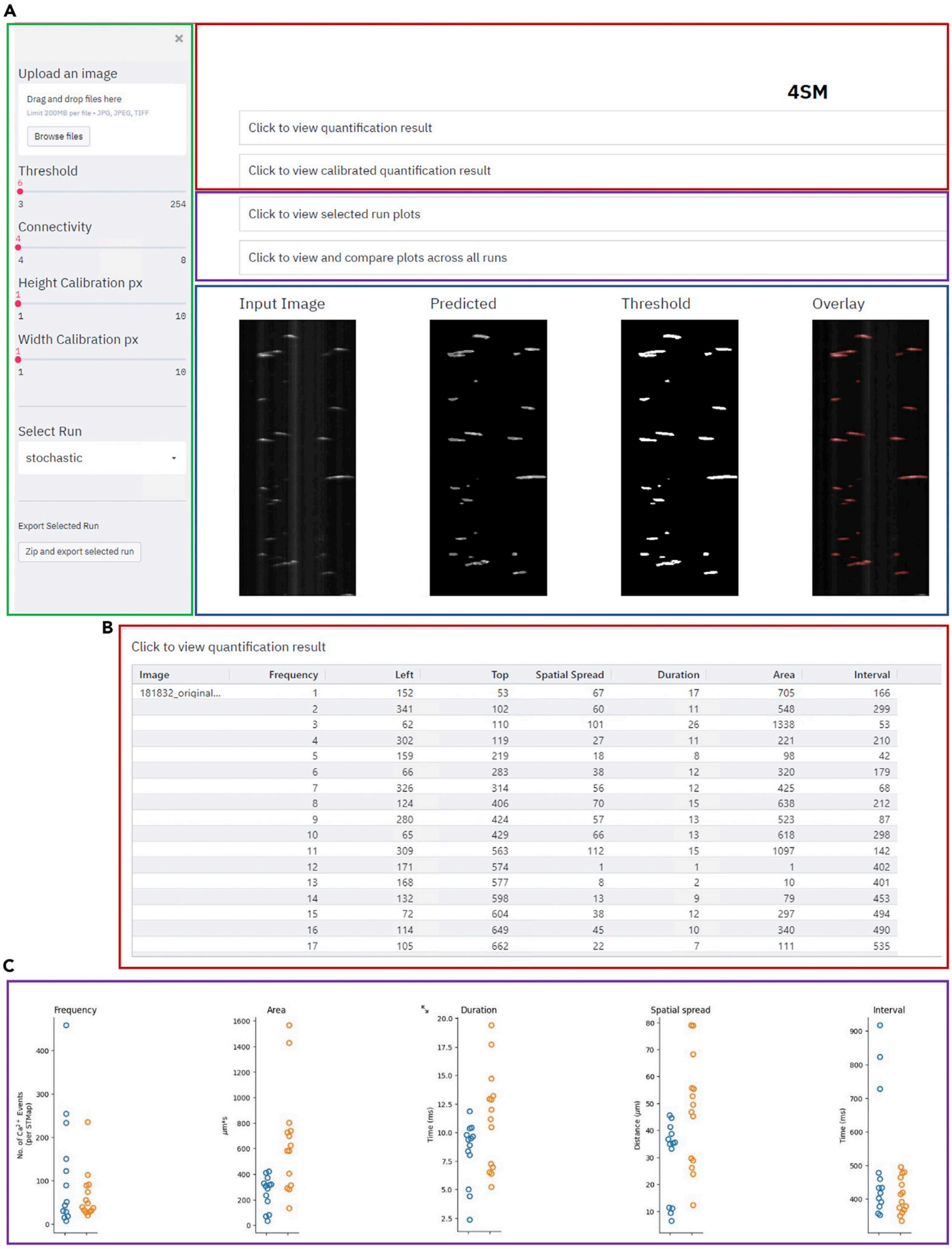
We employed a web-based solution to implement our python code (4SM) to successfully provide a software tool capable of analyzing subcellular fluorescent signals. The software tool includes novel algorithms to effectively analyze fluorescent signal STMaps and can be run through a user-friendly graphical interface as shown in Figure 7. The deep learning algorithms were implemented using Python, and the graphical interface was implemented using Streamlit—an open-source Python library. A key advantage of using Streamlit is to build the software user interface without additional frameworks requirements to be installed. Also, the software can run on any operating system that has Python installed and can be accessible through the web.

4SM is capable of batch processing datasets to allow fast data analysis throughput. The graphical user interface enables the user to specify one or more input images that can be uploaded using drag and drop functions. The tool facilitates a real-time display of the output of the algorithm in a single view and allows for user control of certain input parameters such as threshold, connectivity, pixel calibration, and stride parameters to enhance the degree of event segmentation (Figure 7A). The graphical user interface enables visualization of original, segmented, predicted maps, and overlay images for each input map image. The software provides multiple quantification parameters including the frequency, area, duration, and spatial spread of signal events and enables the user to select, view, and download the output quantification parameters of current or previous runs (Figure 7B).

Within the tool, quantified data can be visualized as graphical plots (Figure 7C) to facilitate the data interpretation on the fly. Each output image and dataset quantification can be visualized and exported within the software.

DISCUSSION

We provide a new software tool ''4SM,'' a spatiotemporal subcellular signal segmentation and analysis tool that provides a seamless workflow and incorporates state-of-the-art machine learning segmentation. The software demonstrates fast segmentation, quantification of events, visualization, and graphical output of data within one tool. Overall, 4SM provides an automated end-to-end solution for subcellular spatiotemporal signal analysis that is both effective and accurate. At present, only a handful of groups have developed methods that use machine learning to segment Ca^{2+} signals (Denis et al., 2020; Giovannucci et al., 2019; Soltanian-Zadeh et al., 2019). Most of these studies do not permit high throughput of large datasets and are limited to analyze specific subsets of cells. They also rely on waveform analysis methods for data retrieval, which typically do not provide a redout of spatial information within a given cell. The major advantages of using STMaps for dynamic fluorescent signal analysis is that they can provide a platform to



Upload an image

Drag and drop files here

Limit 200MB per file • JPG, JPEG, TIFF

Threshold: 6 (3–254)

Connectivity: 4 (4–8)

Height Calibration px: 1 (1–10)

Width Calibration px: 1 (1–10)

Select Run: stochastic

Export Selected Run

Zip and export selected run

Click to view quantification result

Click to view calibrated quantification result

Click to view selected run plots

Click to view and compare plots across all runs

Click to view quantification result

Image	Frequency	Left	Top	Spatial Spread	Duration	Area	Interval
181832_original...	1	152	53	67	17	705	166
	2	341	102	60	11	548	299
	3	62	110	101	26	1338	53
	4	302	119	27	11	221	210
	5	159	219	18	8	98	42
	6	66	283	38	12	320	179
	7	326	314	56	12	425	68
	8	124	406	70	15	638	212
	9	280	424	57	13	523	87
	10	65	429	66	13	618	298
	11	309	563	112	15	1097	142
	12	171	574	1	1	1	402
	13	168	577	8	2	10	401
	14	132	598	13	9	79	453
	15	72	604	38	12	297	494
	16	114	649	45	10	340	490
	17	105	662	22	7	111	535

Figure 7. 4SM interface features

(A) Representative image of the 4SM interface showing the user control segmentation and calibration settings (green box). The STMaps segmentation results are visualized within the tool (blue box). Quantification results (red box) and graphical plots (purple box) control are easily accessible by drop-down menus.

(B) The quantifications of Ca^{2+} signals parameters are displayed numerical format and (C) plotted into graphical plots to aid in interpretation of the data. An example analysis comparison between datasets from various 4SM STMaps that are divided into rhythmic (blue circles) and stochastic (orange circles) groups. n = 17.

analyze a diverse type of cells independently of their shape or origin and can be used to segment and analyze a variety of subcellular dynamic fluorescent signals to monitor ions, pH, protein trafficking, or changes in cellular voltage. Thus, by combining the use of spatiotemporal maps of fluorescent signals and deep learning methodology for segmentation and analysis, we created a universal tool that can be utilized across multiple cell types and can interpret data from a variety of dynamic subcellular fluorescent probes. We validated this point by training our model with two morphologically and functionally distinct cell types of interstitial cells where the 4SM model was effective in both cases.

Very recently, machine learning and deep learning approaches were adopted for Ca^{2+} imaging for STMap analysis by our group. Combining Waikato Environment for Knowledge Analysis (Weka) segmentation, we incorporated fast implementation of fast-random-forest and selective feature learning for Ca^{2+} event segmentation (Leigh et al., 2020). However, this technique requires hand-picked features for learning and segmentation, and user-error diminishes its precision. The 4SM model overcomes these limitations by automating this process using end-to-end deep learning architecture to provide a more robust standardization and high-throughput analysis of cellular Ca^{2+} dynamics.

Our 4SM architecture incorporates multiple attention-based skip connections in generators and comprises novel residual blocks for both generators and discriminators. In addition, it employs reconstruction, feature-matching, and perceptual loss along with adversarial training to fundamentally learn shared features across domains. Because of these features, the 4SM model can distinguish distinct domain-specific features with high precision and robustness for image segmentation analysis.

Cross-domain image translation is widely employed in medical image inpainting for standard statistical analysis (Dalca et al., 2018; Eilertsen et al., 2008; Van Tulder and de Bruijne, 2015) and utilized for converting images of one modality to another interchangeably. However, no technique has been incorporated to learn manifold features with coarse-to-fine generators or multiscale discriminators for coarse and fine feature learning. Our 4SM model successfully combines the learn manifold features utilizing the coarse and fine generators to achieve segmentation accuracy in an effective manner.

Most of the image-to-image translation models are either focused on domain level transformation or combining style and textures of two images. For instance, some models use attention modules to generate high-resolution images to extract local feature information without utilizing perceptual loss (Choi et al., 2020; Kim et al., 2019), whereas other models emphasize more on incorporating perceptual loss with different styles of target images (Park et al., 2019; Wang et al., 2018). To adapt to these changes, our 4SM architecture combines perceptual loss and a multiscale discriminator to retain global information like the spatial spread and duration of the Ca^{2+} transient events. In addition, our model utilizes a feature-matching loss and introduces new multiattention modules to retain local features such as co-localization and boundary of signal events. The visual representation and quantitative results prove that our segmentation technique surpasses the current state-of-the-art architectures.

In conclusion, our new subcellular signal segmenting analysis tool, 4SM, provides an automated, accurate, and fast workflow that can achieve high throughput of large datasets and can be utilized across various fluorescence signaling pattern applications.

Limitations of the study

This study provides a novel solution for the fast and accurate segmentation and analysis of dynamic fluorescence cellular signals. However, the proposed software requires newer GPUs to function properly and run at the intended speed, and many newer GPUs can be costly, which might limit the software's accessibility. This limitation is not specific to 4SM as most deep learning architectures rely on a GPU-based computation. The GPU requirement limits the use of the software in commonly used image analysis software tools such as Image J. Despite these limitations, 4SM software can provide fast and accurate segmentation of cellular fluorescent signals to facilitate seamless signal analysis.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - ○ Lead contact
  - ○ Materials availability
  - ○ Data and code availability

ACKNOWLEDGMENTS

Funding: This project was supported by R01 DK120759 from the NIDDK.

The authors would like to thank Wesley A. Leigh and Guillermo Del Valle for analyzing Ca^{2+} imaging data and conducting imaging experiments.

Grant Support: This project was supported by R01 DK-120759 from the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK).

AUTHOR CONTRIBUTIONS

S.K., K.H., A.T., and S.B designed the experiments. S.K., K.H., A.B., S.R., H.M., and S.B. collected and analyzed the data. S.K., A.B., and S.B. drafted the article. S.K., A.B., H.M., A.T., K.S., and S.B. revised the article for intellectual content.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: January 11, 2022
Revised: April 4, 2022
Accepted: April 18, 2022
Published: May 20, 2022

REFERENCES

Alom, M.Z., Hasan, M., Yakopcic, C., Taha, T.M., and Asari, V.K. (2018). Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. Preprint at arXiv, 1802.06955.

Alom, M.Z., Yakopcic, C., Hasan, M., Taha, T.M., and Asari, V.K. (2019). Recurrent residual U-Net for medical image segmentation. J. Med. Imaging 6, 014006.

Baker, S.A., Drumm, B.T., Cobine, C.A., Keef, K.D., and Sanders, K.M. (2018). Inhibitory neural regulation of the Ca (2+) transients in intramuscular interstitial cells of cajal in the small intestine. Front. Physiol. *9*, 328. https://doi.org/10.3389/fphys.2018.00328.

Baker, S.A., Drumm, B.T., Saur, D., Hennig, G.W., Ward, S.M., and Sanders, K.M. (2016). Spontaneous Ca^{2+} transients in interstitial cells of Cajal located within the deep muscular plexus of the murine small intestine. J. Physiol. *594*, 3317–3338. https://doi.org/10.1113/jp271699.

Baker, S.A., Hennig, G.W., Salter, A.K., Kurahashi, M., Ward, S.M., and Sanders, K.M. (2013). Distribution and Ca^{2+} signalling of fibroblast-like (PDGFRα+) cells in the murine gastric fundus. J. Physiol. *591*, 6193–6208. https://doi.org/10.1113/jphysiol.2013.264747.

Baker, S.A., Hwang, S.J., Blair, P.J., Sireika, C., Wei, L., Ro, S., Ward, S.M., and Sanders, K.M. (2021a). Ca^{2+} transients in ICC-MY define the basis for the dominance of the corpus in gastric pacemaking. Cell Calcium *99*, 102472. https://doi.org/10.1016/j.ceca.2021.102472.

Baker, S.A., Leigh, W.A., Del Valle, G., De Yturriaga, I.F., Ward, S.M., Cobine, C.A., Drumm, B.T., and Sanders, K.M. (2021b). Ca^{2+} signaling driving pacemaker activity in submucosal interstitial cells of Cajal in the murine colon. eLife *10*, e64099. https://doi.org/10.7554/elife.64099.

Becker, C., Rigamonti, R., Lepetit, V., and Fua, P. (2013). Supervised feature learning for curvilinear structure segmentation. In International conference on medical image computing and computer-assisted intervention (Springer), pp. 526–533.

Berridge, M.J., and Dupont, G. (1994). Spatial and temporal signalling by calcium. Curr. Opin. Cell Biol. *6*, 267–274. https://doi.org/10.1016/0955-0674(94)90146-5.

Bradski, G. (2000). The openCV library. Dr. Dobb's J. Softw. Tools Prof. Program. 25, 120–123.

Brown, M., and Lowe, D.G. (2003). October. Recognising panoramas. In ICCV, 3ICCV, p. 1218.

Burt, P.J., and Adelson, E.H. (1987). The Laplacian pyramid as a compact image code. In Readings in Computer Vision (Elsevier), pp. 671–679.

Cai, Y., Landis, M., Laidley, D.T., Kornecki, A., Lum, A., and Li, S. (2016). Multi-modal vertebrae recognition using transformed deep convolution network. Comput. Med. Imag. Graph. *51*, 11–19. https://doi.org/10.1016/j.compmedimag.2016.02.002.

Chartsias, A., Joyce, T., Papanastasiou, G., Semple, S., Williams, M., Newby, D., Dharmakumar, R., and Tsaftaris, S.A. (2018). Factorised spatial representation learning: application in semi-supervised myocardial segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (Springer), pp. 490–498.

Chen, Q., and Koltun, V. (2017). Photographic image synthesis with cascaded refinement networks. In Proceedings of the IEEE international conference on computer vision, pp. 1511–1520.

Chen, W., and Hays, J. (2018). Sketchygan: towards diverse and realistic sketch to image synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9416–9425.

Chen, X., Xu, C., Yang, X., and Tao, D. (2018). Attention-gan for object transfiguration in wild images. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 164–180.

Cheng, H., Lederer, M.R., Lederer, W.J., and Cannell, M.B. (1996). Calcium sparks and [Ca^{2+}] i waves in cardiac myocytes. Am. J. Physiol. Cell Physiol. *270*, C148–C159. https://doi.org/10.1152/ajpcell.1996.270.1.c148.

Choi, Y., Uh, Y., Yoo, J., and Ha, J.-W. (2020). Stargan v2: diverse image synthesis for multiple domains. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8188–8197.

Chollet, F. (2017). Xception: deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251–1258.

Colman, M.A., Pinali, C., Trafford, A.W., Zhang, H., and Kitmitto, A. (2017). A computational model of spatio-temporal cardiac intracellular calcium handling with realistic structure and spatial flux distribution from sarcoplasmic reticulum and t-tubule reconstructions. PLoS Comput. Biol. *13*, e1005714. https://doi.org/10.1371/journal.pcbi.1005714.

Commowick, O., Warfield, S.K., and Malandain, G. (2009). Using Frankenstein's creature paradigm to build a patient specific atlas. In International Conference on Medical Image Computing and Computer-Assisted Intervention (Springer), pp. 993–1000.

Cordier, N., Delingette, H., Lê, M., and Ayache, N. (2016). Extended modality propagation: image synthesis of pathological cases. IEEE Trans. Med. Imag. *35*, 2598–2608. https://doi.org/10.1109/tmi.2016.2589760.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213–3223.

Dalca, A.V., Bouman, K.L., Freeman, W.T., Rost, N.S., Sabuncu, M.R., and Golland, P. (2018). Medical image imputation from image collections. IEEE Trans. Med. Imag. *38*, 504–514. https://doi.org/10.1109/tmi.2018.2866692.

De Vos, B.D., Wolterink, J.M., De Jong, P.A., Viergever, M.A., and Išgum, I. (2016). 2D image classification for 3D anatomy localization: employing deep convolutional neural networks. In Medical Imaging 2016: Image Process., *9784*Medical Imaging 2016: Image Process. (International Society for Optics and Photonics), p. 97841Y.

Dekel, T., Gan, C., Krishnan, D., Liu, C., and Freeman, W.T. (2018). Sparse, smart contours to represent and edit images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3511–3520.

Denis, J., Dard, R.F., Quiroli, E., Cossart, R., and Picardo, M.A. (2020). DeepCINAC: a deep-learning-based Python toolbox for inferring calcium imaging neuronal activity based on movie visualization. Eneuro *7*. ENEURO.0038. https://doi.org/10.1523/eneuro.0038-20.2020.

Denton, E., Chintala, S., Szlam, A., and Fergus, R. (2015). Deep generative image models using a laplacian pyramid of adversarial networks. Preprint at arXiv. https://doi.org/10.48550/arXiv:1506.05751.

Drumm, B.T., Large, R.J., Hollywood, M.A., Thornbury, K.D., Baker, S.A., Harvey, B.J., McHale, N.G., and Sergeant, G.P. (2015). The role of Ca^{2+} influx in spontaneous Ca^{2+} wave propagation in interstitial cells of Cajal from the rabbit urethra. J. Physiol. *593*, 3333–3350. https://doi.org/10.1113/jp270883.

Drumm, B.T., Sergeant, G.P., Hollywood, M.A., Thornbury, K.D., McHale, N.G., and Harvey, B.J. (2014). The role of cAMP dependent protein kinase in modulating spontaneous intracellular Ca(2)(+) waves in interstitial cells of Cajal from the rabbit urethra. Cell Calcium *56*, 181–187. https://doi.org/10.1016/j.ceca.2014.07.002.

Du, J., Li, W., Lu, K., and Xiao, B. (2016). An overview of multi-modal medical image fusion. Neurocomputing *215*, 3–20. https://doi.org/10.1016/j.neucom.2015.07.160.

Eilertsen, K., Nilsen Tor Arne Vestad, L., Geier, O., and Skretting, A. (2008). A simulation of MRI based dose calculations on the basis of radiotherapy planning CT images. Acta Oncol. *47*, 1294–1302. https://doi.org/10.1080/02841860802256426.

Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., and Zisserman, A. (2015). The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vis. *111*, 98–136. https://doi.org/10.1007/s11263-014-0733-5.

Fedigan, S., Bradley, E., Webb, T., Large, R.J., Hollywood, M.A., Thornbury, K.D., McHale, N.G., and Sergeant, G.P. (2017). Effects of new-generation TMEM16A inhibitors on calcium-activated chloride currents in rabbit urethral interstitial cells of Cajal. Pflugers Arch. *469*, 1443–1455. https://doi.org/10.1007/s00424-017-2028-5.

Ganin, Y., and Lempitsky, V. (2014). $$ N^ 4$$-fields: neural network nearest neighbor fields for image transforms. In Asian Conference on Computer Vision (Springer), pp. 536–551.

Giovannucci, A., Friedrich, J., Gunn, P., Kalfon, J., Brown, B.L., Koay, S.A., Taxidis, J., Najafi, F., Gauthier, J.L., Zhou, P., et al. (2019). CaImAn an open-source tool for scalable calcium imaging data analysis. Elife *8*, e38173. https://doi.org/10.7554/elife.38173.

He, L., Chao, Y., Suzuki, K., and Wu, K. (2009). Fast connected-component labeling. Pattern Recogn. *42*, 1977–1987. https://doi.org/10.1016/j.patcog.2008.10.013.

He, Q., Li, X., Kim, D.N., Jia, X., Gu, X., Zhen, X., and Zhou, L. (2020). Feasibility study of a multi-criteria decision-making based hierarchical model for multi-modality feature and multi-classifier fusion: applications in medical prognosis prediction. Inf. Fusion *55*, 207–219. https://doi.org/10.1016/j.inffus.2019.09.001.

Hennig, G.W., Costa, M., Chen, B.N., and Brookes, S.J.H. (1999). Quantitative analysis of peristalsis in the Guinea-pig small intestine using spatio-temporal maps. J. Physiol. *517*, 575–590. https://doi.org/10.1111/j.1469-7793.1999.0575t.x.

Hennig, G.W., Spencer, N.J., Jokela-willis, S., Bayguinov, P.O., Lee, H.t., Ritchie, L.A., Ward, S.M., Smith, T.K., and Sanders, K.M. (2010). ICC-MY coordinate smooth muscle electrical and mechanical activity in the murine small intestine. Neuro Gastroenterol. Motil. *22*, e138–e151. https://doi.org/10.1111/j.1365-2982.2009.01448.x.

Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., and Belongie, S. (2017). Stacked generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5077–5086.

Iglesias, J.E., Konukoglu, E., Zikic, D., Glocker, B., Van Leemput, K., and Fischl, B. (2013). Is synthesizing MRI contrast useful for inter-modality analysis? In International Conference on Medical Image Computing and Computer-Assisted Intervention (Springer), pp. 631–638.

Ioffe, S., and Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning (PMLR), pp. 448–456.

Kamran, S.A., Tavakkoli, A., and Zuckerbrod, S.L. (2020). Improving robustness using joint attention network for detecting retinal degeneration from optical coherence tomography images. In 2020 IEEE International Conference On Image Processing (ICIP) (IEEE), pp. 2476–2480.

Kim, J., Kim, M., Kang, H., and Lee, K. (2019). U-GAT-IT: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. Preprint at arXiv. https://doi.org/10.48550/arXiv:1907.10830.

Kingma, D.P., and Ba, J. (2014). Adam: a method for stochastic optimization. Preprint at arXiv. https://doi.org/10.48550/arXiv:1412.6980.

Lee, H.T., Hennig, G.W., Park, K.J., Bayguinov, P.O., Ward, S.M., Sanders, K.M., and Smith, T.K. (2009). Heterogeneities in ICC Ca2+ activity within canine large intestine. Gastroenterology *136*, 2226–2236. https://doi.org/10.1053/j.gastro.2009.02.060.

Leigh, W.A., Del Valle, G., Kamran, S.A., Drumm, B.T., Tavakkoli, A., Sanders, K.M., and Baker, S.A. (2020). A high throughput machine-learning driven analysis of Ca^{2+} spatio-temporal maps. Cell Calcium *91*, 102260. https://doi.org/10.1016/j.ceca.2020.102260.

Lentle, R.G., and Hulls, C.M. (2018). Quantifying patterns of smooth muscle motility in the gut and other organs with new techniques of video spatiotemporal mapping. Front. Physiol. *9*, 338. https://doi.org/10.3389/fphys.2018.00338.

Li, L., Zhao, X., Lu, W., and Tan, S. (2020). Deep learning for variational multimodality tumor segmentation in PET/CT. Neurocomputing *392*, 277–295. https://doi.org/10.1016/j.neucom.2018.10.099.

Li, R., Zhang, W., Suk, H.-I., Wang, L., Li, J., Shen, D., and Ji, S. (2014). Deep learning based imaging data completion for improved brain disease diagnosis. In International Conference on Medical Image Computing and Computer-Assisted Intervention (Springer), pp. 305–312.

Lim, J.H., and Ye, J.C. (2017). Geometric gan. Preprint at arXiv. https://doi.org/10.48550/arXiv:1705.02894.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C.L. (2014). Microsoft coco: common objects in context. In European Conference on Computer Vision (Springer), pp. 740–755.

Maas, A.L., Hannun, A.Y., and Ng, A.Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In Proc. Icml., *30*Proc. Icml. (Citeseer), p. 3.

Park, T., Liu, M.-Y., Wang, T.-C., and Zhu, J.-Y. (2019). Semantic image synthesis with spatially-adaptive normalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2337–2346.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A.A. (2016). Context encoders: feature learning by inpainting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2536–2544.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation (Springer), pp. 234–241.

Roome, C.J., and Kuhn, B. (2018). Simultaneous dendritic voltage and calcium imaging and somatic recording from Purkinje neurons in awake mice. Nat. Commun. *9*, 3388. https://doi.org/10.1038/s41467-018-05900-3.

Roth, H.R., Lu, L., Liu, J., Yao, J., Seff, A., Cherry, K., Kim, L., and Summers, R.M. (2015). Improving computer-aided detection using convolutional neural networks and random view aggregation. IEEE Trans. Med. Imag. *35*, 1170–1181. https://doi.org/10.1109/tmi.2015.2482920.

Roy, S., Carass, A., and Prince, J. (2011). A compressed sensing approach for MR tissue contrast synthesis. In Biennial International Conference on Information Processing in Medical Imaging (Springer), pp. 371–383.

Sancho, M., Bradley, E., Garcia-Pascual, A., Triguero, D., Thornbury, K.D., Hollywood, M.A., and Sergeant, G.P. (2017). Involvement of cyclic nucleotide-gated channels in spontaneous activity generated in isolated interstitial cells of Cajal from the rabbit urethra. Eur. J. Pharmacol. *814*, 216–225. https://doi.org/10.1016/j.ejphar.2017.08.020.

Sangkloy, P., Lu, J., Fang, C., Yu, F., and Hays, J. (2017). Scribbler: controlling deep image synthesis with sketch and color. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5400–5409.

Sergeant, G.P., Johnston, L., McHale, N.G., Thornbury, K.D., and Hollywood, M.A. (2006). Activation of the cGMP/PKG pathway inhibits electrical activity in rabbit urethral interstitial cells of Cajal by reducing the spatial spread of Ca^{2+} waves. J. Physiol. *574*, 167–181. https://doi.org/10.1113/jphysiol.2006.108621.

Soltanian-Zadeh, S., Sahingur, K., Blau, S., Gong, Y., and Farsiu, S. (2019). Fast and robust active neuron segmentation in two-photon calcium imaging using spatiotemporal deep learning. Proc. Natl. Acad. Sci. U S A *116*, 8554–8563. https://doi.org/10.1073/pnas.1812995116.

Straub, S.V., Giovannucci, D.R., and Yule, D.I. (2000). Calcium wave propagation in pancreatic acinar cells: functional interaction of inositol 1, 4, 5-trisphosphate receptors, ryanodine receptors, and mitochondria. J. Gen. Physiol. *116*, 547–560. https://doi.org/10.1085/jgp.116.4.547.

Teramoto, A., Fujita, H., Yamamuro, O., and Tamaki, T. (2016). Automated detection of pulmonary nodules in PET/CT images: ensemble false-positive reduction using a convolutional neural network technique. Med. Phys. *43*, 2821–2827. https://doi.org/10.1118/1.4948498.

Van Tulder, G., and de Bruijne, M. (2015). Why does synthesized data improve multi-sequence classification? In International Conference on Medical Image Computing and Computer-Assisted Intervention (Springer), pp. 531–538.

Waadt, R., Krebs, M., Kudla, J., and Schumacher, K. (2017). Multiparameter imaging of calcium and abscisic acid and high-resolution quantitative calcium measurements using R-GECO1-mTurquoise in Arabidopsis. New Phytol. *216*, 303–320. https://doi.org/10.1111/nph.14706.

Wang, C., Yang, G., Papanastasiou, G., Tsaftaris, S.A., Newby, D.E., Gray, C., Macnaught, G., and MacGillivray, T.J. (2020a). DiCyc: GAN-based deformation invariant cross-domain information fusion for medical image synthesis. Inf. Fusion *67*, 147–160. https://doi.org/10.1016/j.inffus.2020.10.015.

Wang, K., Zheng, M., Wei, H., Qi, G., and Li, Y. (2020b). Multi-modality medical image fusion using convolutional neural network and contrast pyramid. Sensors *20*, 2169. https://doi.org/10.3390/s20082169.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., and Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8798–8807.

Xian, W., Sangkloy, P., Agrawal, V., Raj, A., Lu, J., Fang, C., Yu, F., and Hays, J. (2018). Texturegan: controlling deep image synthesis with texture patches. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8456–8465.

Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. (2019). Self-attention generative adversarial networks. In International Conference on Machine Learning (PMLR), pp. 7354–7363.

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D.N. (2017). Stackgan: text to photo-realistic image synthesis with stacked generative adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, pp. 5907–5915.

Zhou, T., Thung, K.-H., Liu, M., Shi, F., Zhang, C., and Shen, D. (2020). Multi-modal latent space inducing ensemble SVM classifier for early dementia diagnosis with neuroimaging data. Med. Image Anal. *60*, 101630. https://doi.org/10.1016/j.media.2019.101630.

Zhu, J.-Y., Krähenbühl, P., Shechtman, E., and Efros, A.A. (2016). Generative visual manipulation on the natural image manifold. In European conference on computer vision (Springer), pp. 597–613.

Zhuang, J. (2018). LadderNet: Multi-path networks based on U-Net for medical image segmentation. Preprint at arXiv, 1810.07810.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Experimental models: Organisms/strains		
GCaMP6f-floxed mice (B6;129S-*Gt(ROSA)26Sor*$^{tm95.1(CAG\text{-}GCaMP6f)Hze}$/J)	Jackson Laboratories	Strain #:024105; RRID:IMSR_JAX:024105
Kit-Cre mice (c-Kit$^{+/Cre-ERT2}$)	Provided by Dr. Dieter Saur	https://doi.org/10.1113/JP271699
Software and algorithms		
Fiji, version 2.0.0-rc-69/1.52	NIH	https://fiji.sc/
4SM, Subcellular Fluorescence Analysis Software	This paper	https://doi.org/10.2139/ssrn.4016761; https://github.com/SharifAmit/CalciumGAN/tree/main
STMapAuto, Ca^{2+} Analysis plugin	GitHub	https://doi.org/10.1016/j.ceca.2020.102260

RESOURCE AVAILABILITY

Lead contact

Further information should be directed to and will be fulfilled by the lead contact Dr. Sal Baker (sabubaker@med.unr.edu).

Materials availability

Software code generated in this study have been deposited to Github: https://github.com/SharifAmit/4SM/tree/main and https://doi.org/10.2139/ssrn.4016761.

All data are available in the main text.

Data and code availability

- The original software "4SM" is an open-source and publicly available from https://doi.org/10.2139/ssrn.4016761.
- Or Github: https://github.com/SharifAmit/4SM/tree/main.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

METHOD DETAILS

Tissue preparation

Colonic, antral and small intestinal tissues were removed from animals and small segments (2 cm in length) were incubated in Krebs-Ringer bicarbonate solution (KRB) as previously described (Baker et al., 2016). Tissues were cut along the mesenteric region and contents were removed. The mucosa and submucosal layers were removed and the remaining *tunica muscularis* was placed into a 60 mm Sylgard coated dish and pinned flat.

Ca^{2+} imaging

Muscle sheets isolated from the colon, stomach, and jejunum (~5.0 × 10.0 mm) were pinned down and perfused with 37°C KRB solution. Tissues were equilibration for a period of 1 h. As previously described (Baker et al., 2018); We used a spinning-disk confocal microscope (CSU-W1 spinning disk; Yokogawa Electric Corporation) for all Ca^{2+} imaging experiments. The confocal head is connected to Nikon Eclipse FN1 microscope equipped with a 20 × 0.5 NA, 40 × 0.8 NA, and 60 × 1.0 NA CFI Fluor lens (Nikon instruments INC, NY, USA). Laser at 488 nm wavelength were directed using a Borealis system (ANDOR Technology, Belfast, UK). EMCCD Camera (Andor iXon Ultra; ANDOR Technology, Belfast, UK) was used to capture the GCaMP6f emission. Images were acquired at 33 frames per second using MetaMorph software

(Molecular Devices INC, CA, USA). Nicardipine (100 nM) was used during the imaging experiments to minimize contractile artifacts.

Ca^{2+} event analysis

Analysis and quantifications of Ca^{2+} activity in ICC was performed as previously described (Baker et al., 2016, 2021b). Briefly, movies of Ca^{2+} activity in ICC (30 s long) were converted to a stack of TIFF (tagged image file format) images and imported into Fiji (version 2.0.0-rc-69/1.52, National Institutes of Health, MD, USA, https://fiji.sc/) for analysis. Whole cell ROIs were used to generate spatiotemporal maps (STMaps) of Ca^{2+} activity in individual ICC. STMaps allow for a more complete representation of individual cell Ca^{2+} signals through a 2D image that describes both space of a cell (x axis) and time (y axis) or visa versa. STMaps are a result of spatiotemporal reslicing of an X-Z section in a movie or image stack, where Z is time (Figure 1B). The space of a cell (x axis) is generated in the STMap as result of pixel line or via line-scan average analysis and the pixel values are plotted in time. Each horizontal unit (cell space) in the STMaps represent a pixel or average of pixels in the original image. The coded pixels from each frame of the movie were used to construct a single row and sequential rows were placed under each other to produce an STMap. STMaps presented in the results were generated by rotating image stacks so that ICC were oriented either horizontally or vertically and STMaps of Ca^{2+-}induced fluorescence changes averaged across the diameter of the cell and were constructed using the reslice and z project functions. Ca^{2+} maps area overlap percentages were calculated using Fiji and Pearson's coefficient between original STMaps and segmented 4SM STMaps were obtained using the Just Another colocalization plugin (JACoP; https://imagej.nih.gov/ij/plugins/track/jacop2.html).

EXPERIMENTAL MODEL AND SUBJECT DETAILS

Animals

GCaMP6f-floxed mice (B6;129S-*Gt(ROSA)26Sor*$^{tm95.1(CAG-GCaMP6f)Hze}$/J) were acquired from Jackson Laboratories (Bar Harbor, MN, USA) and crossed with Kit-Cre mice (c-Kit$^{+/Cre-ERT2}$), provided by Dr. Dieter Saur (Technical University Munich, Munich, Germany). Kit-Cre-GCaMP6f mice were injected with tamoxifen at 6–8 weeks of age (2 mg for three consecutive days), as previously described (Baker et al., 2021a, 2021b). 15 days after tamoxifen injection, Kit-Cre-GCaMP6f mice were anaesthetized by isoflurane inhalation (Baxter, Deerfield, IL, USA) and sacrificed by cervical dislocation. All procedures were approved by the Institutional Animal Use and Care Committee at the University of Nevada, Reno. All animals used and the protocols carried out in this study were in accordance with the National Institutes of Health Guide for the Care and Use of Laboratory Animals.

QUANTIFICATION AND STATISTICAL ANALYSIS

Quantification and statistics of Ca^{2+} activity was reported using 4SM, Ca^{2+} event frequency in ICC was expressed as the number of events fired per cell per second (sec^{-1}). The duration of Ca^{2+} events was expressed as full duration (ms), and Ca^{2+} event spatial spread was expressed as μm of cell propagated per Ca^{2+} event. Unless otherwise stated, data is represented as mean ± standard error (S.E.M.). Statistical analysis was performed using either a student's *t*-test or with an ANOVA with a Dunnett post hoc test where appropriate. When describing data throughout the text, n refers to the number of STMaps used in that dataset.

4SM algorithms and implementation

The 4SM tool was developed using Python and the GAN-based algorithm was implemented using Keras (https://keras.io/) and Tensorflow Libraries (https://www.tensorflow.org/). For image pre-processing, visualization, and making utilities for training pipelines we used NumPy (https://numpy.org/), Matplotlib (https://matplotlib.org/), and Pillow (https://pillow.readthedocs.io/en/stable/) packages. Finally, the Streamlit library (https://streamlit.io/) was used to build the user interface for 4SM.

4SM testing

Dataset of 81 STMaps from ICC Ca^{2+} movies were professionally annotated. The dataset was then divided into two groups; group1 of 64 STMaps used for training and cross-validation of 4SM and group 2 of unseen 17 STMaps were used for testing the analysis tool. Coding of 4SM was performed using a PC desktop equipped with dual graphic cards RTX 2080 (Dell 2018; Dell inc; USA).

Download and installation

The open-source tool “4SM” can be downloaded from Github: https://github.com/SharifAmit/4SM/tree/main. Detailed description of installation and downloading the software is provided in the repository (README file) and documentation. A video with captions is also included to provide step by step demonstration of the software installation process (https://www.youtube.com/watch?v=t2LsQkyAGQc). For successful operation in a seamless manner additional components need to be installed which are provided in the repository. The software currently supports 8-bit JPG, PNG and TIF gray-scale images. The minimum hardware requirements for running the software: RAM: 16 GB, GPU: NVIDIA GPU 3 GB or higher memory with CUDA 10.0 support) and CPU Cores: 4 or highe.