Res-CR-Net, a residual network with a novel architecture optimized for the semantic segmentation of microscopy images

Hassan Abdallah 6,9, Brent Formosa 2,4,5,9, Asiri Liyanaarachchi 1,5, Maranda Saigh 4,9, Samantha Silvers 4,9, Suzan Arslanturk 3,2, Douglas J Taatjes 7, Lars Larsson 8, Bhanu P Jena 2,4,5,9 and Domenico L Gatti 1,2

1 Department of Biochemistry, Microbiology and Immunology, Wayne State University, Detroit, MI, United States of America
2 NanoBioScience Institute, Wayne State University, Detroit, MI, United States of America
3 Department of Computer Science, College of Engineering, Wayne State University, Detroit, MI, United States of America
4 Department of Physiology, Wayne State University, Detroit, MI, United States of America
5 Center for Molecular Medicine & Genetics, Wayne State University, Detroit, MI, United States of America
6 Department of Mathematics, Wayne State University, Detroit, MI, United States of America
7 Department of Pathology and Laboratory Medicine, University of Vermont, Burlington, VT, United States of America
8 Karolinska Institute, Stockholm, Sweden
9 contributed equally to the study
E-mail: dgatti@med.wayne.edu

Keywords: deep neural networks, microscopy, segmentation

Abstract
Deep neural networks (DNN) have been widely used to carry out segmentation tasks in both electron microscopy (EM) and light/fluorescence microscopy (LM/FM). Most DNNs developed for this purpose are based on some variation of the encoder-decoder U-Net architecture. Here we show how Res-CR-Net, a new type of fully convolutional neural network that does not adopt a U-Net architecture, excels at segmentation tasks traditionally considered very hard, like recognizing the contours of nuclei, cytoplasm and mitochondria in densely packed cells in either EM or LM/FM images.

1. Introduction
In recent years the task of semantic segmentation has attracted significant interest in biomedical fields in which automated annotation of images is an important process for the extraction of functional information or for 3D reconstruction [1]. Since first introduced, Fully Convolutional Networks (FCN) [2], in which the last fully connected layers are replaced with convolutional layers, have steadily improved the accuracy of semantic segmentation. Additional improvements have been achieved with the introduction of atrous convolutions in DeepLab models [3,4], and the widespread adoption of U-Net models [5–8] with encoder-decoder architecture. A conditional random field (CRF) [9] has been added to some deep neural networks (DNNs) as a post processing step in order to refine the final segmentation [3,4], or embedded in the NN itself in the form of ‘CRF-as-Recurrent Neural Network (RNN)’ [10] for an end-to-end segmentation task [11].

The idea behind the U-Net architecture is that segmentation can be conceptually decomposed in two operations: (1) semantic content extraction in the encoder arm of the neural network (NN), and (2) progressive addition/replacement of the extracted semantic content to the original size image in the decoder arm of the NN. However, it is intuitively hard to understand why these two operations cannot proceed smoothly and progressively in a pixel-wise fashion, without the need for first shrinking and then re-expanding the concept field.

Here, we introduce a type of residual NN [12], Res-CR-Net, that combines residual blocks based on separable, atrous Convolutions (the C in CR), with a final residual block based on RNNs [13] (the R in CR), instead of a CRF, to refine the segmentation task by exploiting the smooth transition between adjacent rows and columns in images. This network displayed excellent performance when tasked to segment images from
either electron or light microscopy in three/four separate categories, using for training only a small number of images with 15-fold augmentation.

2. Methods

2.1. Microscopy

2.1.1. Transmission electron microscopy (EM) of rat liver cells.

Rat liver tissue was prefixed for 24 h in 4% p-formaldehyde (PFA), and then stored in 2% PFA until further processing was performed. Small tissue pieces were further fixed in half-strength Karnovsky’s fixative (1.5% glutaraldehyde, 1.0% formaldehyde in 0.1 M cacodylate buffer) for 80 min at 4 °C, followed by post-fixation for 1 h at 4 °C in 1% OsO₄ in 0.1 M cacodylate buffer. Samples were then dehydrated in a graded series of ethanol concentration, treated with propylene oxide, and embedded in Spurr’s resin. Ultrathin sections were cut with a diamond knife, retrieved onto grids, and contrasted with alcoholic uranyl acetate and lead citrate. Grids were viewed with a JEOL 1400 transmission electron microscope (JEOL USA, Inc. Peabody, MA) operating at 80 kV, and digital images were acquired with an AMT-XR611 11 megapixel CCD camera (Advanced Microscopy Techniques, Danvers, MA).

2.1.2. Fluorescence microscopy (FM) of human skeletal muscle tissue sections.

Human skeletal muscle biopsies were obtained from patients admitted to intensive care unit (ICU). Part of each skeletal muscle biopsy was fixed in 4% PFA, washed in phosphate buffered saline (PBS) prior to setting the tissue in optimal cutting temperature (OCT) compound, frozen, cryo-sectioned at a thickness of 8–10 µm, and placed on Poly-L-Lysine coated glass slides. Tissues sections were first hydrated for 48 h at 4 °C in PBS pH 7.4, then exposed to 10 mM sodium citrate pH 6.0 for 30 min at 60 °C, followed by six washes (2 min per wash) with PBS pH 7.4 at RT. Sections were then blocked with MaxBlock blocking medium (Active Motif) containing 0.1% Tween-20 for 2 h at RT, followed by six washes (2 min per wash) with PBS pH 7.4 containing 0.1% Tween-20 at RT. Tissue sections were finally incubated at 4 °C overnight with myosin heavy chain antibody conjugated to fluorophore [Slow Twitch MYH7-Alexa Flour 647 (red) Santa Cruz Biotechnology, Inc], at 1:50 dilution in PBS pH 7.4, washed twice (5 min per wash) with PBS containing 0.1% Tween-20 at RT, followed by mounting and imaging. In order to label nuclei, sections were incubated with 0.2 µM of 4′,6-diamidino-2-phenylindole (DAPI) (Molecular Probes, Life Technologies, Carlsbad, CA) in PBS pH 7.4, and washed twice (5 min per wash) with PBS containing 0.1% Tween-20 at RT. Imaging was carried out using a Zeiss apotome imager Z1 fluorescence microscope with extended depth of focus, and rendered through Zeiss Imaging Software (Carl Zeiss AG).

2.2. Datasets

2.2.1. Electron microscopy.

Grayscale images (262 × 400 px, reduced from the original 2622 × 4000 px) of rat liver cells were obtained as described in [14]. The segmentation task was to recognize nuclei, mitochondria, and anything else that is neither nuclei nor mitochondria (figures 1(a)–(d)).

In this case, both images and ground truth binary masks were of dimensions (262 × 400 × 1), with three masks (one for each class) per image. Res-CR-Net was trained for 90 epochs with 8 images/8 × 3 masks (corresponding to one batch), and validated against four images/4 × 3 masks.

2.2.2. Fluorescence microscopy.

The dataset consisted of RGB images (300 × 300 px reduced from the original 600 × 600 px) of human skeletal muscle cells from biopsies stained using myosin heavy chain antibody conjugated to a fluorophore. The segmentation task was to recognize the nuclei, the cytoplasm, the boundaries between myofibers, and possible gaps or empty spaces between the myofibers (figure 1(e)–(h)). In this case, color images were of dimensions (300 × 300 × 3), and ground truth binary masks were of dimensions (300 × 300 × 1), with four masks (one for each class) per image. Res-CR-Net was trained for 90 epochs with 10 images/10 × 4 masks (corresponding to one batch), and validated against six images/6 × 4 masks.

2.3. Image pre-processing

2.3.1. Image annotation and ground truth labels.

Ground truth masks for the training and validation sets were obtained using MATLAB® ImageSegmenter (EM images) and ImageLabeler (FM images). These programs only provide a convenient graphical interphase for the hand drawing of regions of interest, and thus the ground truth masks used here were not produced by an algorithm, but created by human annotation.
2.3.2. Data augmentation.
To avoid overfitting and to decrease the time and labor involved in hand labelling the regions of interest in the dataset images, we relied on geometric data augmentation. Each pair of image and ground truth mask(s) was sheered or rotated at random angles, shifted with a random center, vertically or horizontally mirrored, and randomly scaled in/out. The parts of the image left vacant after the transformation were filled in with reflecting padding. During training, each epoch consisted of 15 steps (each step being a batch), and a batch consisted of all images in the dataset. Images in a batch were not shuffled, but each batch of images underwent a different type of augmentation as determined by the consecutive calls of a random number generator starting from a fixed initial seed. The same type of augmentation was applied to an image and its segmentation masks. For example, if the dataset consisted of 10 images and $10 \times (\text{number of classes})$ segmentation masks as labels, each batch contained 10 augmented images and $10 \times (\text{number of classes})$ augmented masks, and an epoch ended when the feeding of 15 different augmented batches to the network was complete. Thus, in every epoch the network trained on 150 different augmented images (out of the original 10), and corresponding $150 \times (\text{number of classes})$ augmented masks as labels, thus achieving 15-fold augmentation/epoch.
Figure 2. Architecture of Res-CR-Net. In an \( n + m \) levels network, the net inputs are batches of images of dimensions [batch size, rows, columns, number of channels], there are 1 STEM BLOCK, \( n-1 \) CONV RES BLOCKS, \( m \) LSTM RES BLOCKS, and the net outputs, matching the labels, are batches of dimensions [batch size, rows, columns, number of classes] (top panel). The middle and bottom panels present a schematic view of a CONV RES BLOCK and a LSTM RES BLOCK, respectively. In this example, the parallel branches of the CONV RES BLOCK feature atrous, separable convolutions with kernels of size [3,3], [5,5], [7,7], and dilation rates [1,1], [3,3], [5,5], respectively. The STEM BLOCK differs from the CONV RES BLOCK only for lacking the initial Batch Normalization and Leaky ReLU operations that are marked with a star symbol in the CONV RES BLOCK. In the LSTM RES BLOCK, the 4D tensor emerging from the previous layer is expanded to a 5D tensor, and the following convolutional LSTM layers processes tensor slices derived either from the 2nd and 4th or from the 3rd and 4th dimension of the input feature map. This operation corresponds to scanning row-by-row both the input map and the map rotated by 90° (which is the same as scanning the input map also column-by-column).

We found this training strategy to be more effective, although slightly slower, than the alternative strategy of using smaller batches, with each epoch processing the entire dataset, and training for a larger number of epochs. It should be noted that this approach is possible in Keras only using the option 'flow from directory', but not with the options 'flow' or 'flow from dataframe'.

2.4. Architecture of Res-CR-Net
A flowchart of the architecture of Res-CR-Net is shown in figure 2. It combines two types of residual blocks, a Convolutional Residual Block (CONV RES BLOCK) and a Long Short Term Memory Residual Block (LSTM RES BLOCK), which are repeated in a linear path along which the dimensions of the intermediate feature maps remain identical to those of the input image and of the output mask(s).
1. **CONV RES BLOCK.** The residual path of this block consists of three parallel branches of separable/atrous convolutions that produce feature maps with the same spatial dimensions as the original image. The kernel sizes and dilation rates used in this block can be customized at will. In this study we have used kernel sizes of [3,3], [5,5], and [7,7], with dilation rates of [1,1], [3,3], and [5,5], respectively.

As stressed in [4], the rationale for using multiple kernel sizes and dilations is to extract object features at various receptive field scales. Res-CR-Net offers the option of concatenating (the default) or adding the parallel branches inside the residual block before adding them to the shortcut connection. A Spatial Dropout layer follows each residual block. A single STEM block, which differs from the CONV RES BLOCK only for lacking the initial Batch Normalization and Leaky ReLU operations (marked with a star symbol in figure 2), processes the initial input. *n*-1 CONV RES BLOCKS can be concatenated. Since the spatial dimensions of the output feature maps remain constant in the chain of repeated blocks, a dense version [15, 16] of the module (which is offered as an option in our deposited code) can be easily constructed without increasing the number of parameters by adding the output of each block to the output of all subsequent blocks.

2. **LSTM RES BLOCK.** This block features a residual path with two orthogonal bidirectional 2D convolutional long short term memory (LSTM) [17] layers processing, respectively, the rows and columns of the input map from previous layers. For this purpose, the 4D tensor emerging from the previous layer first undergoes a virtual dimension expansion to a 5D tensor (i.e. from [4, 260, 400,3], [batch size, rows, columns, number of classes] to [4, 260, 400,3,1], [batch size, rows, columns, number of classes, number of channels]). In this example, the 2D LSTM layer accepts 260 consecutive tensor slices of dimensions [400,3,1] as the input data at each iteration. Each slice is convolved with a filter of kernel size [3,3, number of channels] with 'same' padding, and returns a slice of the same dimension. If more than one filter is used, the 3rd dimension of the resulting slice reflects the number of filters. For example, in one-directional mode the LSTM layer returns a tensor of dimensions [4, 260, 400,3,1 × number of filters]. In bidirectional mode it returns a tensor of dimensions [4, 260, 400,3,2 × number of filters]. In most applications a single filter suffices, but slightly more accurate results can be obtained by increasing the number of filters. The intuition behind using a convolutional LSTM layer for this operation lies in the fact that adjacent image rows share most features, which can be memorized in the LSTM unit. This type of intuition is perhaps somewhat related to the idea of representing the mean-field iterations of a CRF as a recurrent neural network [11]. Since the same intuition applies also to the image columns, in our example the expanded feature map of dimensions [4, 260, 400,3,1] from the earlier part of the network is transposed in the 2nd and 3rd dimension to a tensor of dimensions [4, 400, 260,3,1]. In this case the LSTM layer processes 400 consecutive tensor slices of dimensions [260,3,1] as the input data at each iteration, returning a tensor of dimensions [4, 400, 260,3,2 × number of filters], which is transposed again to [4, 260, 400,3,2 × number of filters]. The two LSTM output tensors are then added and the final dimension (whose size also depends on the number of filters used) is collapsed by summing its elements, leading to a final tensor of dimensions [4, 260, 400,3], which is added to the shortcut path. *m* LSTM RES BLOCKS can be concatenated. While using very few parameters, LSTM blocks are computationally expensive, and thus Res-CR-Net allows for not including them in the network architecture, if in a particular case they do not improve the segmentation accuracy.

A Leaky ReLU activation is used throughout Res-CR-Net. After the last residual block a softmax activation layer is used to project the feature map into the desired segmentation. The Dice coefficient, \( D \) [18–22], and the Dice loss, \( L_D \), defined as:

\[
L_D(\hat{y}, y) = 1 - D = 1 - \frac{2 \sum_i \hat{y}_i y_i + s}{\sum_i \hat{y}_i + \sum_i y_i + s}
\]

with \( \hat{y} \equiv \{\hat{y}_i\}, \hat{y}_i \in [0,1] \) being the probabilities for the \( i \)-th pixel, \( y \equiv \{y_i\}, y_i \in \{0,1\} \) being the corresponding ground truth labels, and \( s \) a smoothing scalar, have been shown to increase performance over the cross entropy loss for semantic segmentation tasks [23], and thus are a popular choice among practitioners. In [8] it was found empirically that loss functions of the Dice class containing squares in the denominator behave better in pointing to the ground truth, and that faster training convergence can be obtained by complementing the loss with a dual form that measures the overlap area of the complement of the regions of interest. These observations, and additional theoretical considerations [8] led to the implementation of the Tanimoto loss, defined as:

\[
L_T(\hat{y}, y) = 1 - T(\hat{y}, y)
\]
Table 1. Performance metrics for Res-CR-Net with the EM and FM validation datasets.

| Image Size (px) | aDice | aPrecision | aRecall | aF1 |
|-----------------|-------|------------|---------|-----|
| EMgs            | 262 × 400 | 0.922      | 0.924   | 0.921 | 0.922 |
| FMgb            | 300 × 300 | 0.906      | 0.908   | 0.904 | 0.906 |
| FMps            | 300 × 300 | 0.878      | 0.882   | 0.874 | 0.878 |

aDice = \( \frac{2 \times TP + FP \times FN}{TP + FP + FN + TP} \)
aPrecision = \( \frac{TP}{TP + FP} \)
aRecall = \( \frac{TP}{TP + FN} \)
aF1 score = \( \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \) with TP = true positive, FP = false positive, FN = false negative.

where \( \tilde{T}(\hat{y}, y) \) is the Tanimoto coefficient with complement:

\[
\tilde{T}(\hat{y}, y) = \frac{T(\hat{y}, y) + T(1 - \hat{y}, 1 - y)}{2}
\]

with \( T(\hat{y}, y) \) defined as:

\[
T(\hat{y}, y) = \frac{\sum (\hat{y}_i y_i + s)}{\sum (\hat{y}_i^2 + y_i^2)} - \frac{\sum \hat{y}_i y_i + s}{\sum \hat{y}_i y_i + s}
\]

While in this study we have used the Tanimoto loss function throughout, Res-CR-Net can use a variety of other losses, including categorical crossentropy, dice loss [18–22], categorical crossentropy + dice loss, Tanimoto loss without complement, categorical crossentropy + Tanimoto loss, all appropriately weighted to account for class imbalance. Weights are derived either with a contour aware scheme, by replacing a step-shaped cutoff at the edges of the mask foreground with a raised border that separates touching objects of the same or different classes [6], or with the inverse volume scheme [21], \( w_I = V_I^{-2} \), where \( w_I \) are the weights and \( V_I \) is the total sum of true positives per class I, or with both. Regardless of the loss function used for training, in this study we used the unweighted Dice coefficient defined above as the primary metric to evaluate segmentation accuracy.

2.5. Software

Res-CR-Net was implemented using Keras [24, 25] deep learning library running on top of TensorFlow 2.1 [26], and is publicly available at https://github.com/dgattiwsu/Res-CR-Net. Training and testing were conducted on the Wayne State University high performance computing grid equipped with Intel CPUs and Tesla V100-SXM2-16GB GPUs. Res-CR-Net can process images of any size, without constrains imposed by the down-pooling and up-sampling operations of an encoder-decoder architecture. Three types of masks can be used: (1) a single 3 or 4 channels binary mask/image is suitable to process up to 4 segmentations classes. (2) a single 1 channel (grayscale) mask/image can be used to process an arbitrary number of classes. In this case, the mask must be thresholded with different gray levels (i.e. a mask with three classes would have the regions corresponding to the three categories thresholded at 0,128,255 values). This type of mask is first converted to sparse categorical with each gray level corresponding to a different index (i.e. \([0, 128, 255]\) goes to \([0, 1, 2]\)). Then, pixels identified by indices are converted to one-hot vectors. (3) multiple 1 channel binary masks per image, with a separate mask for each class can also be used to process an arbitrary number of classes. In this case, the masks must be placed in different directories corresponding to the different classes.

3. Results

3.1. EM dataset

When trained for 90 epochs on the EM dataset of 8 grayscale images (262 × 400 px) with 15-fold augmentation (see above), Res-CR-Net (configured with six CONV RES BLOCKS and two LSTM RES BLOCKS) achieved ~90% segmentation accuracy (Dice coefficient: 92.2%, table 1) on the four images of the validation set (figure 3).

The segmentation task in these images was to identify the regions occupied by the cell nuclei, the cytoplasm, and the mitochondria. Thus, three binary segmentation masks were associated with each grayscale image, and the network trained in each epoch on 120 different augmented images and corresponding 120 × 3 augmented binary masks as labels. A block diagram of the network architecture used for this task is shown in figure S1 (available online at stacks.iop.org/MLST/1/045004/mmedia). The total number of parameters refined was 58 978, the memory use was 9.79 GB/(batch of eight images).
Figure 3. Nuclei and mitochondria segmentation in EM images. Electron micrographs of rat liver cells from the validation set with superimposed Res-CR-Net predicted masks for nuclei (cyan, left panel) and mitochondria (violet, right panel). The cytoplasm is identified as all the regions that are neither nuclei nor mitochondria.

Figure 4. Training and validation weighted Tanimoto Loss and Tanimoto coefficient vs. epochs for Res-CR-Net processing of FM images of human muscle cell from biopsies. The total number of refined parameters in this model was 59,165, the memory use was 10.58 GB/(batch of 10 images).

It is worth noting that segmentation of mitochondria in electron micrographs from focused ion beam scanning electron microscopy (FIB-SEM) or automated tape-collecting ultramicrotome scanning electron microscopy (ATUM-SEM) has been the focus of multiple reports [27–31] in recent years. However, due to the variety of mitochondrial structures, as well as the presence of noise, artifacts, and other sub-cellular structures, mitochondria segmentation in EM has proven to be a difficult and challenging task. As apparent
Figure 5. Segmentation of human muscle cells with Res-CR-Net trained on three channel (rgb) color images. Left column: fluorescence micrographs of human skeletal biopsy sections stained for myosin heavy chain. Center column: ground truth masks (nuclei, black; cytoplasm, dark gray; cell boundaries, light gray; gaps/empty spaces, white). Right column: Res-CR-Net predicted masks.

from figure 3, Res-CR-Net performs very well in this task, despite the small number of images used for training.

3.2. FM dataset
When trained for 90 epochs on the FM dataset of 10 images with 15-fold augmentation (see above), Res-CR-Net (configured with six CONV RES BLOCKS and one LSTM RES BLOCK) showed excellent refinement properties as attested by its convergence speed (~30 epochs) and stability (up to 90 epochs) with respect to both the training and validation set (figure 4).

The segmentation task in these images was to identify the cell nuclei, the cytoplasm, the boundaries between muscle cells, and regions that do not correspond to any of the previous three categories. Thus, four binary segmentation masks were associated with each color image, and the network trained in each epoch on 150 different augmented images and corresponding $150 \times 4$ augmented binary masks as labels. In this task
Figure 6. Segmentation of human muscle cells with Res-CR-Net trained on grayscale images. A grayscale micrograph (top left) corresponding to the color micrograph in the 4th row of figure 5, is shown with superimposed predicted masks for nuclei (violet, top right), cytoplasm (yellow, bottom left), and cell boundaries (cyan, bottom right).

Figure 7. Res-U-Net with LSTM block (Res-UR-Net). A 4-level Res-U-Net (panel a) with residual blocks containing 2 parallel branches of separable convolutions with kernels size [3,3] and [5,5] (panel b) is the precursor of Res-CR-Net. The 3-class segmentation performance of this U-Net on some of the color images of the FM dataset is shown in panel c. Left column: fluorescence micrographs of human skeletal biopsy sections stained for myosin heavy chain. Center column: ground truth masks (cytoplasm, red; cell boundaries, green; gaps/empty spaces, blue). Right column: predicted masks.

Res-CR-Net also achieved ~90% segmentation accuracy (Dice coefficient: 90.6%, table 1) on the six images of the validation set (figure 5). A block diagram of the network architecture used for this task is shown in figure S2, Supporting Information.
Table 2. Segmentation metrics for the FM validation datasets using color (rgb) images of different size with a 7-level Res-CR-Net, (as in table 1), or with a 4-level Res-UR-Net of similar performance and execution time. As both networks are fully convolutional, the number of parameters is independent from the image size, but the number of math operations and thus the memory used and the training times depend on it. Data is missing for the processing of 300 × 300 px images by Res-UR-Net, because this network can only process images of dimensions that are powers of 2.

| Image size (px) | Parameters | Memory GB/(batch) | Training time (s/batch) | Dice | Precision | Recall | F1 |
|-----------------|------------|-------------------|------------------------|------|-----------|--------|----|
| Res-CR-Net 256 × 256 | 59 165 | 7.61 | ~8 | 0.900 | 0.907 | 0.894 | 0.900 |
| Res-UR-Net 256 × 256 | 748 634 | 3.414 | ~8 | 0.893 | 0.898 | 0.889 | 0.893 |
| Res-CR-Net 300 × 300 | 59 165 | 10.584 | ~9 | 0.910 | 0.913 | 0.907 | 0.910 |
| Res-UR-Net 300 × 300 | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. | n.a. |
| Res-CR-Net 512 × 512 | 59 165 | 30.823 | ~18 | 0.905 | 0.908 | 0.902 | 0.905 |
| Res-UR-Net 512 × 512 | 748 634 | 13.573 | ~18 | 0.900 | 0.901 | 0.897 | 0.899 |

*1 batch = 10 images

Res-CR-Net identified very accurately the positions and shapes of nuclei (DAPI stained in blue in the leftmost column, and as black spots in the center and rightmost columns of figure 5), clearly distinguishing them from other bright spots in the cell boundary regions, which are due instead to high local concentration of myosin. To determine whether nuclei recognition was somehow facilitated by the presence of a DAPI stained channel in the RGB images used for training, we have retrained Res-CR-Net after first converting all color images to grayscale. In this case, while Res-CR-Net showed somewhat decreased per pixel accuracy on the validation set (Dice coefficient: 87.8%, see also table 1), it was still able to identify correctly all the regions of interest (figure 6).

3.3. Res-CR-Net versus Res-U-Net

The performance of Res-CR-Net with the EM and FM datasets is summarized in table 1 with respect to the metrics (Dice coefficient, Precision, Recall, F1 score) most often used to evaluate neural networks performance.

However, prior to developing the intuition about the Res-CR-Net architecture we experimented with the traditional U-Net architecture. In fact, the LSTM RES BLOCK was initially implemented as the final block of a residual U-Net (for this reason called a Res-UR-Net) with four levels in the encoding arm and four levels in the decoding arm, and tested with some of the color images in the FM dataset on a simpler segmentation task involving only three classes (figure 7).

The residual blocks of Res-UR-Net were designed using the principles outlined in [7, 8], with two parallel convolutional branches of separable convolutions with kernel sizes [3,3], [5,5], respectively. A block diagram of Res-UR-Net for images of 512 × 512 px is provided in figure S3, Supporting Information. The depth of Res-UR-Net (four levels in both the encoder and decoder) is the same as that of the original U-Net [5, 6], slightly more than the three levels of Deep Res-U-Net [7], and slightly less than the six levels of Res-U-Net-a [8]. Thus, Res-UR-Net is in the middle of the range of depths that has been implemented in various flavors of U-Net style architectures. Several Res-U-Net architectures with or without a final LSTM block are also available for comparative testing in the Github deposition of Res-CR-Net (see above).
Using the complete FM dataset, we have now carried out a systematic comparison (table 2) of the Res-CR-Net used in this study against our own Res-UR-Net, trained under the exact same conditions (epoch number, batch size, augmentation parameters, loss, number of processors). From the results presented in table 2 it appears that Res-CR-Net achieves a level of segmentation accuracy that is at least as good if not slightly better than that achieved by Res-UR-Net, despite the latter using approximately 13 times more parameters. Just as noted in the case of Res-CR-Net, the training history of Res-UR-Net shows no significant overfitting (figure 8) of the training set vs. the validation set.

4. Conclusions

In this report we present Res-CR-Net, a neural network featuring a novel FCN architecture that departs significantly from the popular U-Net organization and from the encoder-decoder paradigm. Res-CR-Net shows very good performance in multiclass segmentation tasks of both electron (grayscale, 1 channel) and light microscopy (color, 3 channels, or grayscale, 1 channel) images of relevance in the analysis of pathology specimens. The network was effective in achieving a semantic segmentation of the validation set images into 3 (EM set) or 4 (FM set) classes that was almost indistinguishable from the ground truth (figure 5).

The Res-CR-Net architecture offers some advantages with respect to an encoder-decoder architecture, as its layers contain no pooling or up-sampling operations, and therefore the spatial dimensions of the feature maps at each layer remain unchanged with respect to those of the input images and of the segmentation masks used as labels or predicted by the network. For this reason, Res-CR-Net is completely modular, with residual blocks that can be proliferated in a straight down linear fashion as needed (figure 2), and it can process images of any size and shape without changing layers size and operations. Res-CR-Net also feature a novel type of residual LSTM block, in which two orthogonal convolutional LSTM layers process independently the rows and columns of the feature map emerging from the previous layers. We have found that addition of this block to the network can produce an additional improvement of the final segmentation masks, an effect comparable to that achieved by the CRF post-refinement in other types of networks [10, 11]. One advantage of the LSTM block is, however, its capacity to process batches with multiple images, while the CRF layer is limited in its current implementation to process only one image at a time.

The significantly smaller number of parameters in Res-CR-Net vs. Res-UR-Net is likely to reduce the risk of overfitting the training set, while achieving comparable segmentation accuracy. However, this smaller parameter load does not translate in faster execution times, as Res-CR-Net and Res-UR-Net run at approximately the same speed (table 2). Finally, the benefits provided by the modular architecture of Res-CR-Net come at a price, as Res-CR-Net is more memory hungry than Res-UR-Net, using slightly more than twice the amount of GPU on-board memory per batch of images (table 2).

In this study we have chosen to pursue segmentation tasks that are usually considered particularly difficult, like identifying the envelope of mitochondria in EM images (figure 3), or the cell boundaries in light microscopy fields in which cells are packed at high density (figures 5, 6). Res-CR-Net was particularly effective in both segmentation tasks despite the fact that the training sets consisted of only 8–10 images. On this basis, Res-CR-Net may be most useful in those cases in which the labelling of ground truth classes is laborious, and thus the number of annotated/labelled images in the training set is small.

Acknowledgments

This study was supported in part by the National Science Foundation Grant Nos. EB00303, CBET1066661 (BPJ), and the Erling-Persson Family Foundation, the Swedish Research Council Grant No 8651, and the Stockholm City Council Grant Nos. Alf 20150423 and 20170133 (LL).

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/dgattiwsu/Res-CR-Net.

Authors Contribution

HA, SA and DLG developed the architecture of Res-CR-Net. MS and SS produced all ground truth masks. AL, DJT, LL, BPJ provided the image datasets. HA, LL, SA, BPJ and DLG wrote the manuscript. All authors participated in discussions and proofreading the manuscript.
Software

Source code for Res-CR-Net is deposited at: https://github.com/dgattiwsu/Res-CR-Net

ORCID ID

Domenico L Gatti https://orcid.org/0000-0002-6357-3530

References

[1] Garcia-Garcia A, Orts S, Oprea S, Villena Martinez V and Rodriguez J 2018 A review on deep learning techniques applied to semantic segmentation Appl. Soft Comput. 70 41–65
[2] Long J, Shelhamer E and Darrell T 2015 Fully convolutional networks for semantic segmentation The IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) pp 3431–40
[3] Chen L-C, Papandreou G, Kokkinos I, Murphy K and Yuille A L 2016 DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs arXiv:1606.00915
[4] Chen L-C, Papandreou G, Schroff F and Adam A 2017 Rethinking atrous convolution for semantic image segmentation arXiv:1706.05387
[5] Ronneberger O, Fischer P and Brox T 2015 U-net: convolutional networks for biomedical image segmentation MICCAI (Lecture Notes in Computer Science) vol 9351 (Berlin: Springer) 234–41
[6] Falk T et al 2019 U-Net: deep learning for cell counting, detection, and morphometry Nat. Methods 16 67–70
[7] Zhang Z, Liu Q and Wang Y 2018 Road extraction by deep residual U-net using deep neural network for segmentation in chest radiographs arXiv:1804.05055
[8] Diakogiannis F, I, Waldner F, Caccetta P and Wu C 2019 ResUNet-a: a deep learning framework for semantic segmentation of remotely sensed data ISPRS J. Photogram. 162 94–114
[9] Kroghenbuhl P and Koltun V 2012 Efficient inference in fully connected CRFs with Gaussian edge potentials arXiv:1210.5644
[10] Zheng S et al 2015 Conditional random fields as recurrent neural networks Int. Conf. on Computer Vision (ICCV) (https://doi.org/10.1109/ICCV.2015.179)
[11] Arnab A et al 2018 Conditional random fields meet deep neural networks for semantic segmentation: combining probabilistic graphical models with deep learning for structured prediction IEEE Signal Process. Mag. 35 37–52
[12] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image classification The IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) pp 770–8
[13] Goodfellow I, Bengio Y and Courville A 2016 Deep Learning (Cambridge, MA: MIT Press)
[14] Pernal S P et al 2020 Nanoscale imaging using differential expansion microscopy Histochem. Cell Biol. 153 469–80
[15] Huang S Z, Li X Y, Jiang Z Q, Guo X Q and Men A D 2018 Fully convolutional network with densely feature fusion models for object detection 2018 IEEE Int. Conf. on Multimedia & Expo Workshops (ICMMEW) (San Diego, CA) pp. 1–6
[16] Huang G, Liu Z, van der Maaten L and Weinberger K Q 2016 Densely connected convolutional networks arXiv:1608.06993
[17] Hochreiter S and Schmidhuber J 1997 Long short-term memory Neural Comput. 9 1735–80
[18] Dice L R 1945 Measures of the amount of ecologic association between species Ecology 26 297–302
[19] Sørensen T 1948 A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons Biol. Skr. 5 1–34
[20] Mallett F, Navab N and Ahmadi S-A 2016 Fully convolutional neural networks for volumetric medical image segmentation arXiv:1606.04797
[21] Crum W R, Camara O and Hill D L G 2006 Generalized overlap measures for evaluation and validation in medical image analysis IEEE Trans. Med. Imaging 25 1451–61
[22] Sudre C H, Li W, Vercauteren T, Ourselin S and Cardoso M J 2017 Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations Lecture Notes in Computer Science vol 10553 (Berlin: Springer) (https://doi.org/10.1007/978-3-319-67588-9_28)
[23] Novikov A A, Denis D, Major D, Hladívková J, Wimmer M and Bühler K 2017 Fully convolutional architectures for multi-class segmentation in chest radiographs arXiv:1710.08816
[24] Chollet F 2018 Deep Learning with Python (Shelter Island, NY: Manning Publications Co.) pp 11964
[25] Chollet F Keras (available at: https://keras.io)
[26] Abadi M et al TensorFlow: Large-scale machine learning on heterogeneous systems (available at: https://www.tensorflow.org/)
[27] Li W, Liu J, Xiao C, Deng H, Xie Q and Han H 2018 A fast forward 3D connection algorithm for mitochondria and synapse segmentations from serial EM images BioData Min. 11 24
[28] Casser V, Kang K, Pfister H and Hachem D 2018 Fast mitochondria segmentation for connectomics arXiv:1812.06024
[29] Liu J, Li W, Xiao C, Hong B, Xie Q and Han H 2018 Automatic detection and segmentation of mitochondria from SEM images using deep neural network 2018 40th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC) (18–21 July 2018) pp. 628–31
[30] Xiao C et al 2018 Automatic mitochondria segmentation for EM data using a 3D supervised convolutional network Front. Neuroanat. 12 1–92
[31] Oztel I, Yolcu G, Ersoy I, White T and Bunyak F 2017 Mitochondria segmentation in electron microscopy volumes using deep convolutional neural network 2017 IEEE Int. Conf. on Bioinformatics and Biomedicine pp 1195–200