Transferrable Deep Learning Algorithms for Identifying Area At Risk in Acute Myocardial Infarction

Christopher Collazo[2], Brendon Cara[1], Carla J. Weinheimer[3], Ryan Grabau[1], Dmitry Goldgof[2], Lawrence Hall[2], Samuel Wickline[1], Hua Pan[1]
Colleges of Medicine[1] and Engineering[2], Univ. of South Florida, Tampa, FL; Washington Univ.[3], St. Louis, MO

Introduction: Objective analysis of complex histopathology images with minimized observer bias is critical to many areas of biological research and medical diagnosis. Current automated methods for imaging, processing, and statistical analysis of tissue sections can be sensitive to variability in tissue images, such as color and brightness variations, blur, dust, and tissue folds that create excessive noise and degrade the analyses of microscopic specimens.

Deep learning presents an opportunity for a solution that is insensitive to such noise, adaptable to different tissue types, and able to handle much larger quantities of tissue data than the gold standard manual evaluations done by an expert.

Here, we apply deep learning methods, using deep convolutional neural networks, to sections of heart tissues that have undergone myocardial ischemic injury in order to delineate the areas at risk of injury from normal regions in a fully automated and objective procedure.

Materials and Methodology

Data Set and Imaging: Ten C57BL6 mice underwent left anterior descending (LAD) coronary artery ligation for 90 min, followed by reperfusion for 24 hours. Two fluorescent nanoparticles at different particle sizes of 55nm (magenta: mild) or 250 nm (red: severe) were injected right before the reperfusion to define mild and severe damage, respectively, in the heart. Following excision 24 hours after reperfusion, the heart was perfused with saline to remove blood in the heart, the LAD coronary artery was reoccluded; Lectin-488 (a vascular endothelial marker) was perfused to label blood vessels located exclusively in the nonischemic (no damage) region. This results in four channels for each heart: A “blue” channel (nuclear stain) defining tissue area, a “green” channel indicating nonischemic vascularized tissue, a “red” channel for severe damage, and a “magenta” channel for mild damage. These full-heart images, taken at 162.5 nm/pixel of resolution, are downscaled by a factor of 0.3, resulting in a resolution of 541.67 nm/pixel to be fed to the deep learning model.

The ten resulting microscopy images were hand-segmented by an expert to obtain “ground truth” for the nonischemic (normal), severe, mild, and area at risk (AAR) by “hand-painting” (in GIMP) the zones for each tissue class such that any given pixel in the whole heart image must belong to one of these classes, or a fifth “none” class indicating it is outside of the tissue region.

The deep learning model is translating a four-channel fluorescence input into a five-class segmentation map. Thus, the four fluorescence channels were stacked together, while the ground truth segmentation maps were split into five binary masks, one for each class, which were then stacked together. Each of the ten heart images was then divided into a grid of 512x512 tiles, averaging 300-500 tiles per heart image; across all ten hearts, this yields 4073 tiles. Thus, our final input tile shape is 512x512x4, and our final output (and target) tile shape is 512x512x5.

Convolutional Neural Network Architecture: The raw fluorescence tiles were trained with their corresponding ground truth segmentation tiles in a deep neural network based on the U-net encoder-decoder architecture [1] using convolutional layers [2] to extract information about texture and shape. The architecture features two legs, a central component between them, and a finishing component.

The first leg is an encode path, featuring four down-sampling components. Each consists of two convolutional layers followed by a max-pooling layer [3]. The convolutions in each component increase in number (16, 32, 64, and 128), and the outputs of these convolutions are forwarded to the decode leg.

The central component is a latent feature extractor consisting of two very deep convolutional layers of 256 convolutions modified with a dropout rate of 0.5 [4]. The output of this feature extractor is a latent feature tensor of shape 32x32x256; that is, 256 images of shape 32x32, each a downsampled version of the input image with extracted features made prominent.

The second leg is a decode path consisting of four components, each with an initial up-sample (an inverse of the max-pooling operation) followed by three convolutional layers of decreasing size (128, 64, 32, and 16, mirroring the encode path). The inputs to these convolutions are concatenations of the previous component's upsampling and the outputs from the encode leg.

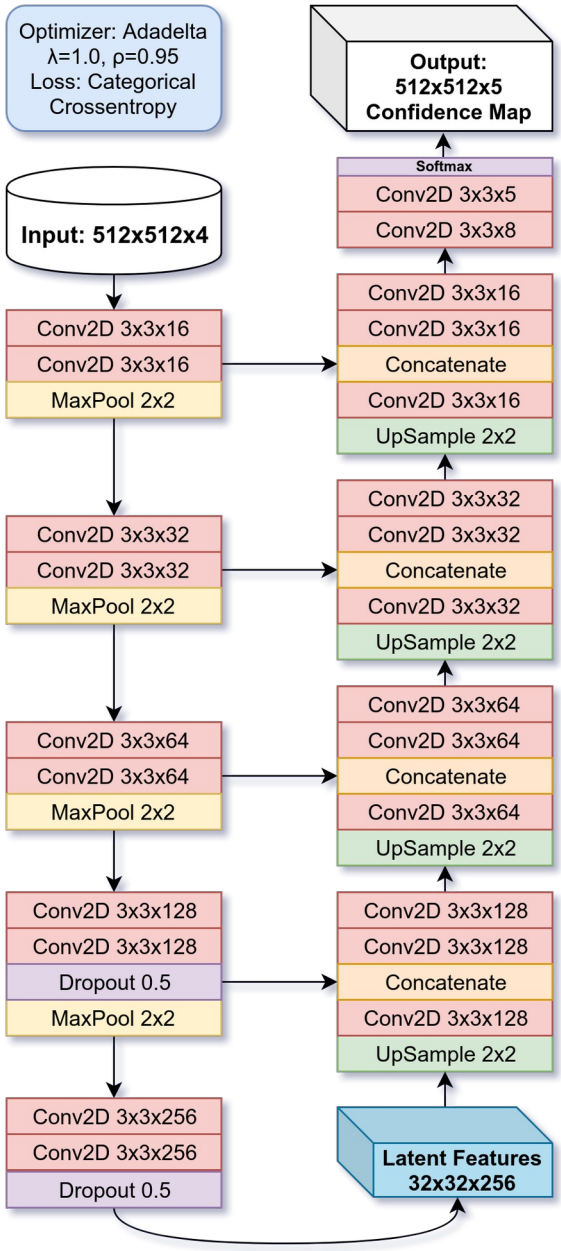

Figure 1: U-Net style convolutional neural network architecture used.

Lastly, there is a finalization component consisting of two convolutional layers which ensures that we obtain the proper number of segmentation class outputs. This output stack is put through a Softmax function [3] to generate a 512x512x5 confidence map; that is, at each pixel location, there are five values representing the degree of confidence that the pixel belongs to the given tissue class. These values add up to 1.0, and thus are relative confidence levels (or "pseudo-probabilities").

The model does gradient descent [3] using the Adadelta algorithm [5], with λ=1.0 and ρ=0.95, with gradient norm clipping of 1.0 to prevent an exploding gradient. The vanishing gradient problem is prevented by the residual paths between the legs of the U in the architecture. Loss is calculated using categorical crossentropy [3].

Training and Testing: The model was trained and tested using the "leave-one-out" cross-validation methodology. This allows us to verify that the model is achieving similar performance across permutations of the training set and test set by training a separate model for each heart (for ten total models), which allows us to single out hearts that show significant variations in metrics and determine why that might be the case. Two hearts are put aside for the validation set, which acts as a barometer of the model's efficacy during training by detecting whether overfitting is occurring, and can be used to stop training early.

While metrics during training and validation are tracked using only individual tile samples, stitching these together during testing creates some amount of noise at the tile borders in the final heart image due to the lack of context past the tile edges, leaving a visible grid pattern. To combat this, we only take the 256x256x5 center of the output tile, throwing away the 128 pixel buffer zone on all sides as trim; this requires us to slide our 512x512 window a smaller distance than otherwise. The result is a very consistent full-heart segmentation map. Final performance metrics were defined by comparing this full-heart segmentation to the full-heart ground truth.

Results and Discussion: The model displays outstanding results on the test hearts, with an average accuracy at the pixel level of 0.9037 across full-heart images. Closer inspection reveals that in several of the hearts, the few accuracy issues are mainly caused by two things: First, the relative ambiguity of the areas surrounding the borders between tissue classes; and second, excessive noise outside of the tissue area, due to particularly intense smears or particle spills.

Both of these are issues best solved by simply adding more data – the more the model trains on borders, the better it will be at discerning them, and the more the model trains on noise objects, the better it will become at identifying and suppressing them.

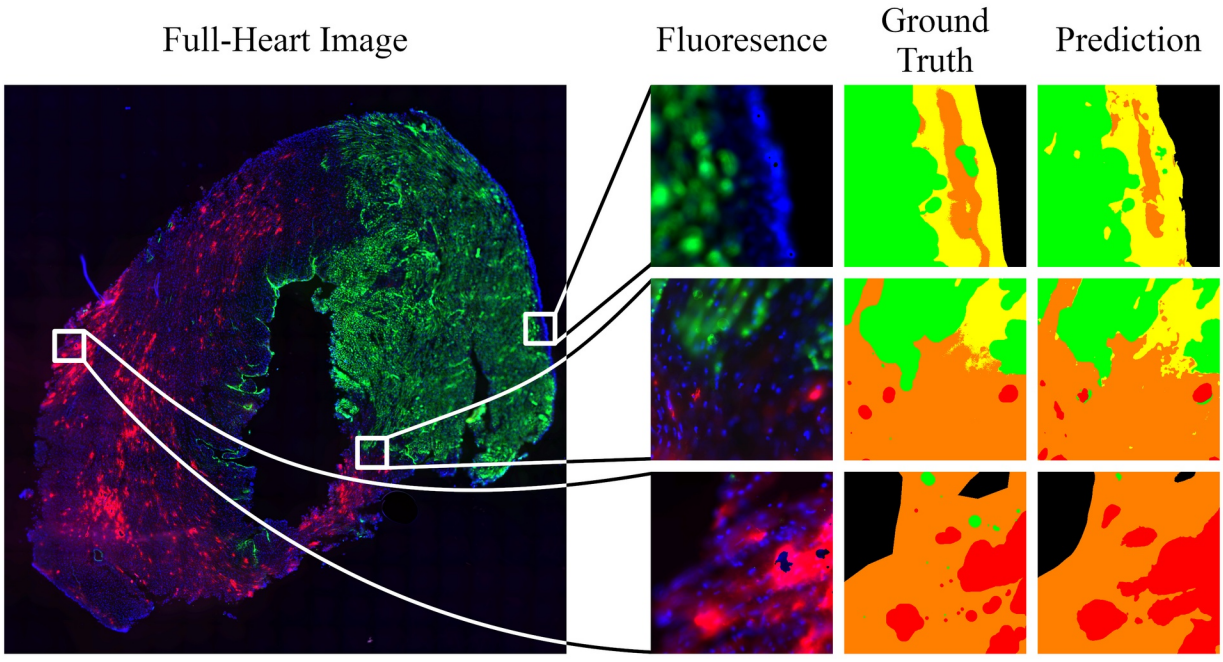

Figure 2: Visual demonstration of model performance on 3 samples from a heart. Class colors are as follows: Black is background, green is normal, yellow is AAR, orange is mild damage, red is severe damage.

We also have analyzed the running time of the model on a single heart. A rough TensorFlow/Python implementation running on four Nvidia GeForce GTX 1080 Ti's finishes segmenting a whole heart in an average 5.15 minutes. A single GTX 970 can segment a whole heart in about 20 minutes. With optimization and utilization of TensorFlow tools which enable faster runtimes, we could expect a production machine to segment in 10 minutes or less, with times only decreasing as computing hardware improves.

Conclusions and Future Work: We have demonstrated that a U-net style encoder-decoder convolutional neural network architecture trained on fluorescence nanoparticle images is effective for automatic segmention of heart tissue according to the severity of myocardial injury. Moreover, the algorithm completes in a timely manner without user interaction. Future work will test newer types of neural network architectures on more extensive training sets to achieve a more comprehensive coverage of unknown data.

Citations

1. Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

2. LeCun, Yann, and Yoshua Bengio. "Convolutional networks for images, speech, and time series." The handbook of brain theory and neural networks 3361.10 (1995): 1995.

3. Goodfellow, Ian, et al. Deep Learning. MIT Press, 2017.

4. Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15.1 (2014): 1929-1958.

5. Zeiler, Matthew D. "ADADELTA: an adaptive learning rate method." arXiv preprint arXiv:1212.5701 (2012).