

Ultrasound Nerve Segmentation – Group 9

Tanner Henry
T2henry@eng.ucsd.edu

Neil Vranicar
Nvranica@eng.ucsd.edu

Stephen West
swest@eng.ucsd.edu

Abstract

After surgery, patients are typically prescribed opioids in order to alleviate pain and aid in the recovery process. Opioids, however, have many adverse health effects such as nausea/vomiting, liver damage, and physical dependence [1]. Alternative solutions for pain mitigation exist but require radiologists to locate nerve clusters in ultrasound images. Our goal is to be able to accurately locate these nerve clusters using a convolutional neural network so that this alternate pain management solution can be accessible without needing a radiologist to locate nerve clusters.

1. Introduction

To accurately identify nerve structures in ultrasound images is no simple task, it requires the availability of highly trained medical specialists. These medical specialists are few and far between and identifying the location of the nerve structures is the key component to inserting Halyard Health’s pain management solution, an indwelling catheter. In order to introduce an indwelling catheter as a replacement to opioids the scarcity of radiologists to locate nerves in ultrasounds must be solved to bring this product to market. Fortunately, identifying an object and its specific location from pre-labeled data is precisely the job for semantic image segmentation models. In this paper, convolutional neural networks will be discussed to solve the problem of locating nerve clusters in ultrasound images.

1.1. Related Work

In D. Ciaran et al. [2], a model to segment neuronal membranes is implemented in a way to allow for localization while extracting features. In order to provide localization, the model is based on a sliding window filter, aiming to predict the class label of each pixel by providing an area to that pixel. Not only does this provide localization, it also increases the dataset size.

In O. Ronneberger et al. [3], a convolutional neural network is introduced for the application of segmenting biomedical images called U-net which expanded on the work by D. Ciaran et al. [2]. The creation of U-net stemmed from the problem that traditional neural networks face of needing very large datasets to train their models.

Within biomedical imaging, these large datasets are very difficult to find due to the expertise needed to create them. This creates a need for a network architecture which can be trained on a relatively small dataset. In U-net, they execute substantial data augmentation, specifically elastic deformations. They keep localization by combining features in the contracting path with the up-sampled output [3]. U-net also builds off the first work to train a fully convolutional network for pixel-wise prediction [4]

In a tutorial posted on the Kaggle [5] dataset, an individual by the username jocicmarko lays out a basic strategy for implementing U-net to this application. He starts with converting the images to numpy binary files to speed up loading later. He pre-processes the data by resizing the image and then feeds it into a customized U-net. This is the basic skeleton that we started our investigation from [6].

With regards to this dataset and competition, many teams experimented with the size of the image. One group found that the larger image sizes produced slightly better scores. This group used horizontal and vertical flips, rotations, shears, and zooms as part of their data augmentation process. They trained the models with Adam over 100-150 epochs and found that there seemed to be a relationship between score and file size. They did some post processing, converting intensities greater than 0.5 to 1 and comparing predicted mask size to the smallest mask size in the training data in order to identify false masks [7].

2. Dataset and features

The dataset consists of ultrasound images and corresponding mask images of the brachial plexus nerve. The dataset was obtained from Halyard Health and consists of 5635 420x580 training ultrasound images from a total of 47 patients and 5508 420x580 test ultrasound images from an unknown number of patients [5]. Sample images from this dataset are shown in Figure 1.

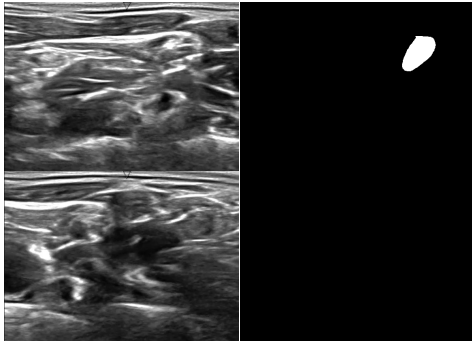


Figure 1: Sample Ultrasound Mask Pairs

This dataset consists of ultrasounds where the brachial plexus is present and also ultrasounds where it is not. In ultrasounds where there is not brachial plexus our model should predict no mask.

Since this dataset was part of a competition the test images are not labeled. To overcome this challenge, the training dataset is split into training, validation, and test sets. This further reduced the size of our dataset and the need for data augmentation grows.

Due to the way the acquisition machine generates image frames from the ultrasound video, there are nearly identical images in the dataset. With these nearly identical images, there are occurrences when the radiologist assigned contradicting masks. As a preprocessing technique, nearly identical images that did not contain a mask were removed. Locating nearly identical images was accomplished by taking the cosine difference between all of the pairs of images in the data set. Images were classified as nearly identical if one minus the cosine distance between them was less than $8e-3$. We then checked to see if the nearly identical images had contradicting masks by taking the dice coefficients between the masks. If the dice coefficient between the masks was less than 0.2, the images had contradicting masks and the image mask pair that had no mask was discarded. This choice of removing the empty mask stems from trusting that the radiologist only labels the image when they are sure there is a mask. Figure 2 shows samples of nearly identical images with contradicting masks with the original ultrasound on the left and the ultrasound with the mask overlay on the right.

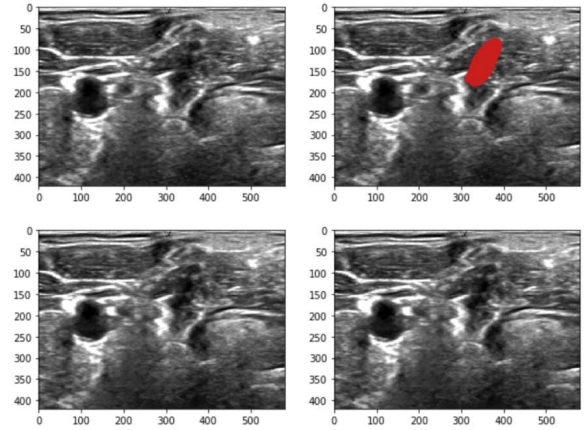


Figure 2: Duplicate Images with Contradicting Masks

Since our model requires more images than available, the training and validation set were augmented by a factor of 3 by randomly rotating each image in the dataset and by flipping each image along the x and y direction. The final size of our training, test, and validation sets are 9012 training images, 2056 validation images, and 2818 testing images.

To train our model the training images are resized to 96x96. This proved to be a good size by comparing results from mini-batches of training sets with differing image sizes. This image size minimized training time while not significantly affecting the model score.

3. Method

Due to the lack of training data for such a specific problem, the Unet architecture is proposed. The benefit of this architecture for biomedical image segmentation is that it makes use of a relatively small set of labeled images and relies on data augmentation to improve the training process. The architecture consists of two paths, a contracting and symmetric expanding. In the contracting path, convolutions followed by an activation and max pooling operations are performed to learn the context in the image. At each layer in the contracting path, the image height and width are cut in half while the depth is doubled. At this stage the location information is lost. In the symmetric expanding path this problem is solved. The use of up-sampling and skip connections from the contracting layer allows for precise localization. Up-sampling is performed through transpose convolutions, doubling the width and height of the image at each layer while cutting the depth in half. At each layer, the transposed convolution is concatenated with the output of the corresponding contracting layer. This concatenation is followed by a convolution, returning the image to its original size. A block diagram of the U-net architecture is shown in Figure 3.

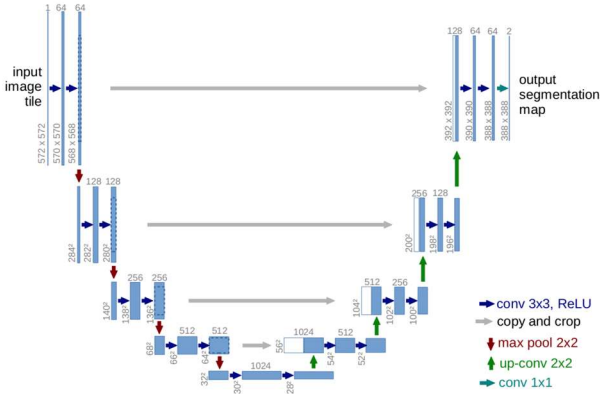


Figure 3: U-net Architecture (image courtesy [3])

At each layer prior to the output, exponential linear unit (ELU) activation is used. A visual representation of this function is shown in Figure 4. ELU activation smooths the input slowly versus its rectified linear unit (RELU) counterparts sharp smoothing.

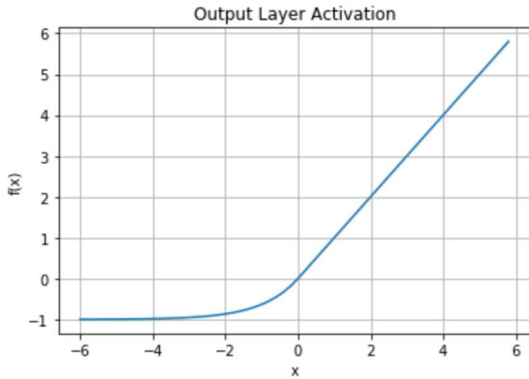


Figure 4: Elu Activation

At the final layer, sigmoid activation is utilized. A visual representation of this function is shown in Figure 5. The sigmoid function takes an input and maps it to the range $[0,1]$.

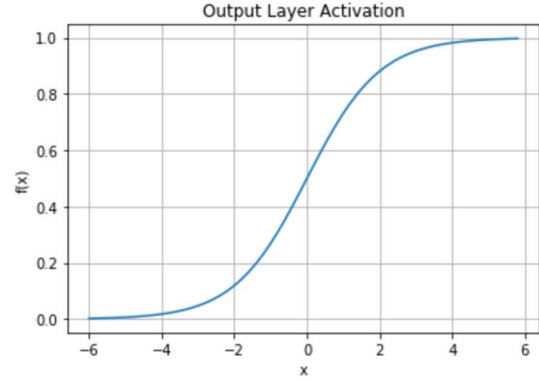


Figure 5: Sigmoid Activation

In our method, post-processing techniques are utilized to further improve the model. The first technique is the threshold for class classification of the output. The output of the model is an image of probabilities that a nerve is located at that pixel location. If this probability is greater or equal to 0.6 the pixel value is mapped to a 1 representing the nerve class; if it is less than 0.6 it is mapped to a 0 representing not a nerve class. The second technique implemented is zeroing out output masks that aren't feasible based on the training mask pixel area statistics. This allows for the reduction of false positives.

4. Results

The metric used to determine the performance of our algorithm is the dice-coefficient in L. Dice [8]. The dice-coefficient is a pixel-wise agreement between two images with a max value of 1 corresponding to images which are the same. The formula for the dice-coefficient is described in equation (1). Furthermore, the model was trained on minimizing the dice-loss which is the negative of the dice-coefficient.

(1)

$$DC = \frac{2|X \cap Y|}{|X| + |Y|}$$

In this project, a few different models were implemented: resnet18 [9], inceptionv3 [10], and U-net. In our implementation of inceptionv3 and resnet18, the models seemed to be overfitting to the training data. Furthermore, the output of these models predicted extremely poorly when no nerve is present in the image. This led us to use the U-net model for the rest of the experimentation.

Within U-net, we found that ELU activation performed better than the proposed RELU activation in the U-net architecture.

The next parameter that improved performance is the threshold for class classification. As stated earlier, the value that provided the best result was 0.6.

Lastly, the size to zero out masks at the output was chosen to be the average mask size in the training set minus 500. This value performed well, however, this may be improved by obtaining a distribution of mask sizes in the training set and using a threshold equal to one standard deviation from the mean.

A visual result of outputs of our model may be viewed in Figure 6. When there is a nerve present in the ultrasound our model performs extremely well (high dice coefficient); however, if there is no nerve present our model doesn't always predict correctly. The middle pair is an example of a mask that can be zeroed out by refining the threshold of feasibility in our post-processing technique 2. However, there are some occasions where our model simply provides us with a poor prediction which is undesirable (bottom pair), and these occurrences shall be minimized in further implementation.

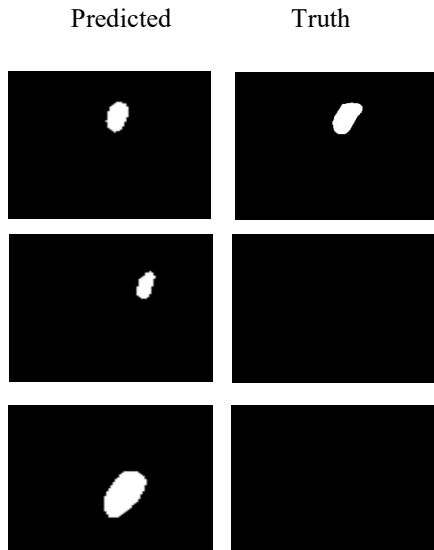


Figure 6 Desirable result (top), result that may be filtered with refined post-processing (middle), and undesirable result (bottom)

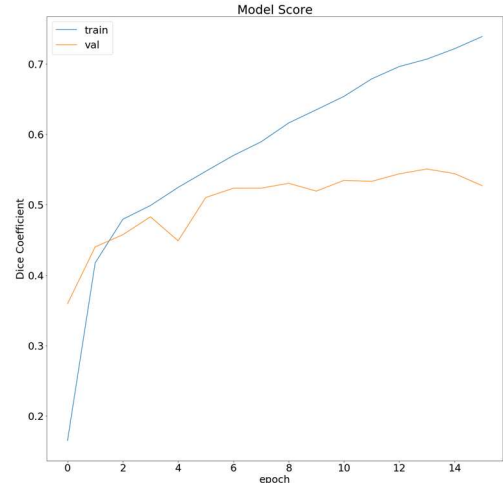


Figure 7: Score at Training Stage

5-fold verification was used to verify the accuracy of this model. A summary of this experiment is represented in Table 1. We chose not to include more folds in our verification because we found that the variance in scores between the 5 folds was relatively low, meaning that including more folds in verification would not significantly increase the accuracy of our score.

	Metrics
1st-fold	0.712
2nd-fold	0.676
3rd-fold	0.706
4th-fold	0.694
5th-fold	0.655
Variance	4e-4
Mean	0.689

Table 1: 5-Fold Verification

From the classification output of our model, the confusion matrix for pixel classification of the nerve class is described in Table 2. A key point to note is that our model was able to minimize false positives well. This is extremely important because if our model provided a medical professional administering the pain management solution with a false positive, the patient may be harmed or experience unnecessary pain. The high rate of false negatives is likely due to human error in labeling the dataset.

	Predicted: Negative	Predicted: Positive
Actual: Negative	TN = 99.62%	FP = 0.38%
Actual: Positive	FN = 44.56%	TP = 55.44%

Table 2: Pixel-wise Confusion Matrix

5. Conclusion and Future Work

The project hinged heavily on our pre-processing algorithm, U-net implementation, and post processing algorithm. Eliminating duplicate ultrasounds with contradicting masks and augmenting the data were critical for our model. U-net implementation with sigmoid activation function on the final layer provided the best results over softmax. Finally, eliminating predicted masks that were smaller than any provided in the training data improved our score as well.

In the future with more time and resources, we would work to expand the network and dataset to allow for classification of a wider variety of nerve clusters. This will allow broader implementation of Halyard Health's pain management solution. We would also like to apply the model to video stream input rather than a single image to allow for real time image labeling; which is essential to allowing a medical professional to accurately insert the catheter during the procedure.

6. Contributions

All of our work was done in a collaborative setting at scheduled times in group study rooms throughout the quarter. Methods, procedures, and techniques were discussed and debated collaboratively. While code was written on individual group member's computers debugging and algorithm authorship was done as a group.

References

- [1] S. A. Schug, D. Zech and S. Grond, "Adverse Effects of Systemic Opioid Analgesics," *Drug Safety*, vol. 7, no. 3, pp. 200-213, 1992.
- [2] D. Ciresan, L. Gambardella, A. Giusti and J. Schmidhuber, "Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images," Semantic Scholars, 2012.
- [3] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," University of Freiburg, Freiburg, Germany, 2015.
- [4] J. Long, E. Shelhamer and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," UC Berkeley, Berkeley, CA, 2015.
- [5] Halyard Health, "Ultrasound Nerve Segmentation," 6 May 2016. [Online]. Available: <https://www.kaggle.com/c/ultrasound-nerve-segmentation/data>. [Accessed 2 May 2019].
- [6] jocimarko, "Github," 4 February 2017. [Online]. Available: <https://github.com/jocimarko/ultrasound-nerve-segmentation>. [Accessed 11 June 2019].
- [7] A. Beam, "Andrew L. Beam: Machine Learning and Medicine," 20 August 2016. [Online]. Available: <http://beamandrew.github.io/deeplearning/2016/08/20/kaggle-segmentation.html>. [Accessed 11 June 2019].
- [8] L. R. Dice, "Measures of the Amount of Ecologic Association Between Species," *Ecology*, vol. 26, no. 3, pp. 297-302, 1945.
- [9] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 10 December 2015. [Online]. Available: <https://arxiv.org/pdf/1512.03385.pdf>.
- [10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *CPVR*, Las Vegas, 201.