Automated Skin Biopsy Analysis with Limited Data

Yung-Chieh Chan, Jerry Zhang(✉), Katie Frizzi, Nigel Calcutt, and Garrison Cottrell

University of California, San Diego, USA
{ychan,jcz001}@ucsd.edu, gary@eng.ucsd.edu

Abstract. In patients with diabetic and other peripheral neuropathies, the number of nerve fibers that originate in the dermis and cross the dermal-epidermal boundary is an important metric for diagnosis of early small fiber neuropathy and determination of the efficacy of interventions that promote nerve regeneration. To aid in the time-consuming and often variable process of manually counting these measurements, we propose an end-to-end fully automated method to count dermal-epidermal boundary nerve crossings. Working with images of skin biopsies immunostained to identify peripheral nerves using current standard operating procedures, we used image segmentation neural networks to distinguish between the dermis and epidermis and an edge detection neural network to identify nerves. We then applied an unsupervised clustering algorithm to identify nerve crossings, producing an automated count. Since our dataset is very small—containing less than one hundred images—we use pretrained models in combination with several image augmentation methods to improve performance on training and inference. The model learns from a human expert's training data better than a human trained by the same expert.

1 Introduction

Automated systems for aiding clinical diagnoses and treatment research have been long sought after both to increase the speed of procedures as well as offer consistent quantification. In particular, semantic segmentation finds applications in many parts of the clinical process.

For the specific challenge of detecting neuropathy, there are already several methods that can automatically identify nerve-like structures. Al-Fahdawi et al. [7] showed that this task can be automated using image preprocessing and edge detection on corneal images. More recent work using deep learning shows improvements over these traditional methods [13]. In particular, several groups have applied U-Nets, an architecture designed for semantic segmentation, to skin biopsies, exactly the problem we approach here [2,10]. In [2], the authors additionally use U-Nets for tracing nerves in skin biopsies. However, in order for the U-Net to work, they enhanced the manually-traced nerves in the training set images with a 6-pixel-wide boundary in order to make the problem simpler for the U-Net. In our work,

G. Zamzmi et al. (Eds.): MILLanD 2022, LNCS 13559, pp. 229–238, 2022.
https://doi.org/10.1007/978-3-031-16760-7_22

we found that U-Nets perform less well than using a state-of-the-art edge detector for nerve tracing, and no enhancement of the nerve traces was necessary.

We build off of these previous approaches using a similar deep learning semantic segmentation method. However, to overcome the problem of very limited data, we employ further methods to ensure that the model will have good performance on unseen data.

In addition, to achieve a fully automated system for detecting neuropathy and measuring nerve growth in images of skin biopsies, we factor the task into two sub-tasks: identifying nerve fibers in the skin, and identifying the dermal-epidermal boundary. For each sub-task, we employ a specialized model, and we combine their results using an unsupervised method to obtain the final nerve crossing annotations.

2 Methods

2.1 Dataset

The dataset consists of 94 images of skin biopsies collected from 13 HIV-infected patients and 11 control participants as part of a study on HIV infection and neuropathy in the United States. Each 1600 × 1200 image contains a portion of the biopsy at 20× magnification. Many focal layers are flattened into a single image to capture depth. A human expert traced the nerves and the boundary between the dermis and epidermis.

Given the limited number of training examples, we perform the following random data augmentations during training: horizontal and vertical translation from –100 to 100 pixels, rotation from –10 to 10°, shearing from –5 to 5°, orientation (the image can be rotated to 0, 90, 180, or 270°), cropping, and horizontal and vertical flipping. For cropping, 800 × 800 patches are taken from each 1600 × 1200 image; this size generates different images with each application of the augmentation algorithm while still retaining enough of the dermis and epidermis to be useful for the model. Similarly, the other applied image augmentations ensure that each training example has an essentially unique configuration of pixels, while retaining vital information about skin structure and not distorting the image beyond what an expert technician would normally be able to work with.

The augmentation algorithm is applied as the model is training such that a random set of parameters is generated and applied for each transformation. As a result, the model is trained on a unique version of every image with each iteration. In combination with this online approach, the wide array of random image augmentation methods is one of the primary ways that we improve the model's generalization and prevent over-fitting.

2.2 Nerve Labeling

In our dataset, nerve labels are about three pixels wide and constitute only a very small portion of the image. Conventional semantic segmentation models

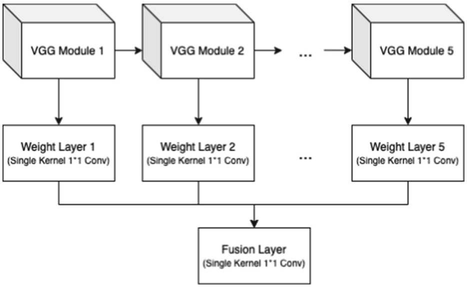

VGG Module 1
VGG Module 2
...
VGG Module 5
Weight Layer 1 (Single Kernel 1*1 Conv)
Weight Layer 2 (Single Kernel 1*1 Conv)
...
Weight Layer 5 (Single Kernel 1*1 Conv)
Fusion Layer (Single Kernel 1*1 Conv)

Fig. 1. Architecture of holistically-nested edge detection (HED)

suffer when there is an extreme class imbalance, failing to capture pixel-level details. After experimenting with various models, we decided to approach the task with an edge detection model.

Architecture. We use the Holistically-Nested Edge Detection (HED) [12] system. A simple illustration of the HED architecture is shown in Fig. 1. HED is a feed-forward deep convolutional neural network based on VGGNet [11]. The model consists of a series of VGG modules with an increasing number of kernels. The input image is processed by these modules sequentially and the multi-channel output of each VGG module is compressed to a single-channel side-output by a convolutional layer with a 1 × 1 kernel. These 1 × 1 convolution layers also serve as weights of the side-outputs. Each side-output focuses on a different scale of edges as deeper VGG modules, with larger receptive fields, capture larger edges. The final edge map is generated by fusing the side-outputs by one 1 × 1 convolution layer.

Loss Function. The loss function we used to train our model is Dice Loss, a commonly used loss for segmentation. Let P be the two-dimensional model output and T be the ground truth. The equation for Dice Loss is

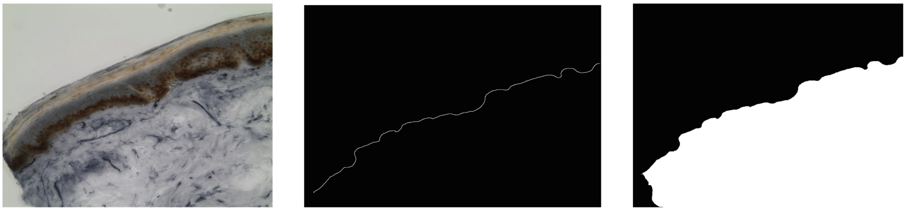
$$L_{Dice}(P,T) = 1 - \frac{2 * \sum_{i,j} P_{ij}T_{ij} + 1}{\sum_{i,j} P_{ij} + \sum_{i,j} T_{ij} + 1}$$

Training Procedure. Due to the scarcity of labeled data and the sparsity of learning signals in each image, we perform transfer learning to improve the model's performance and reduce training time. We adopted the HED model pretrained on BSDS500 dataset [1] from pytorch-hed [9] and fine-tune it on our dataset. The pretrained model was already able to identify the majority of the nerves but also labeled other non-nerve edges. Hence fine-tuning on our data corrected this. We set the learning rate to 10^{-4} and a weight decay penalty of $L_2 = 10^{-3}$, determined by grid search cross-validation. The weight decay parameter is especially important in our case in order to minimize over-fitting on our small dataset. We trained the model for 200 epochs with a learning rate scheduler, which decreases the learning rate when the loss flattens.

2.3 Dermis-Epidermis Boundary Detection

As is the case with the segmentation of nerve fibers, labeling the boundary line between the dermis and epidermis can be very difficult due to class imbalance. However, this boundary is not as easily detected as nerve fibers, since the boundary is not as clearly defined.

To circumvent this problem, we reformulate the labeling scheme: instead of labeling the boundary between the dermis and epidermis, we only task the model with labeling the dermis. From this label, a boundary line between the dermis and epidermis can be generated using standard image processing techniques. In order to transform the labeling scheme, we manually extend the line in the given label to surround the whole dermis, then flood-fill the dermis region. An example is shown in Fig. 2.



(a) Input Image (b) Label (line scheme) (c) Label (dermis scheme)

Fig. 2. An example label for the dermis

Therefore, using the transformed data, the task can be formulated as binary classification between dermis and non-dermis regions, a straightforward task for semantic segmentation, and the categories are relatively balanced. We use a DeepLabV3 model, a recent state-of-the-art approach for semantic segmentation [4]. In particular, we use a pre-trained instance of the model with a ResNet-101 [8] backbone, which we then fine-tune on our dataset.

Architecture. The model is a deep residual convolutional neural network with atrous (also referred to as dilated) convolutions, a key feature that allows the model to have a wider receptive field in the later layers without sacrificing feature map resolution, which is essential for semantic segmentation [4]. Atrous spatial pyramid pooling, a method similar to spatial pyramid pooling but using filters with various atrous rates, further improves the model's capacity to process both global contexts as well as small-scale detail.

Loss Function. We use the standard binary cross-entropy loss function evaluated on a per-pixel basis at the output for training the model on the dermis. To evaluate performance, we use the Dice coefficient, which is discussed in more detail in subsequent sections.

Training Procedure. Since the number of training examples is small, we leverage a version of the model that is pre-trained on PASCAL Visual Object Classes dataset [6], a set of natural images with 20 categories.

From grid search using cross-validation over learning rate and L_2 penalty, we use an initial learning rate of 3×10^{-4} and an L_2 weight decay penalty of 10^{-3}, which again helps to minimize over-fitting. The model is trained on the labeled dermis data using binary cross-entropy loss for 300 iterations, using the same learning rate schedule as above. For fine-tuning, we wrap the model with an initial and final convolutional layer and train the whole model end-to-end on the dataset. In this case, the initial convolutional layer has 3 input channels and 3 output channels, with a kernel size of 3 and padding width of 1 to maintain the resolution of the input. The output convolutional layer uses a kernel size of 1 with 21 input channels and 1 output channel, in order to reduce the dimensionality of the pretrained model's output.

2.4 Nerve Crossing Identification

Our approach for counting nerve crossings consists of three steps: transforming dermis label to dermis-epidermis boundary, clustering intersections, and filtering out invalid crossings. All parameters in this section were optimized by running the same process on the ground truth data and the model output on the same images, and maximizing their consistency.

Transform Dermis Label to Boundary. To obtain a boundary line for the segmented dermis label, we first smooth the boundary by applying a Gaussian filter with $\sigma = 10$ pixels onto the dermis map and binarizing it with a threshold of 0.6. Then, we scan the map for enclosed areas smaller than 0.2 of the image in the dermis map and flip the label for all pixels in that area. This step removes noisy patches from the model's prediction within the dermis. Finally, we extract the dermis-epidermis boundary by selecting the pixels close to the edge of the dermis area with a controllable boundary width. An example is shown in Fig. 3.

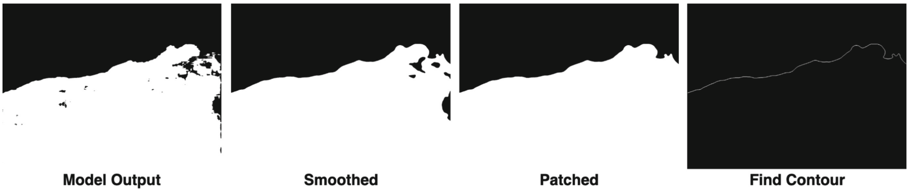


Fig. 3. An example of transforming dermis label to boundary line

Cluster Intersections. In this step, we extract the overlapping pixels of the filtered nerve labels and the dermis-epidermis boundary. The coordinates of the overlapping pixels is then clustered by Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [5] with $\epsilon = 3$ and the minimum number of points in a cluster set to 4. Each cluster is considered a crossing, and the coordinate of each crossing is calculated by averaging all pixels coordinated in the corresponding cluster.

Filter Crossings. For each crossing, we count the number of nerve pixels in a 17 × 17 pixel area centered at the crossing. We filtered out crossings that have less than 5 nerve pixels on both sides of the boundary. This step filters out nerve fibers that do not cross into the dermis but border it.

3 Experimental Setup

3.1 Evaluating the Nerve Tracing Model

The first evaluation metric we used is Dice score. For each image with P as the set of nerve pixels predicted by our model and T as the set of nerve pixels labeled by a human expert, the Dice score is calculated as:

$$Dice(P, T) = \frac{|P \cap T|}{(|P| + |T|)/2}$$

However, this metric is not very effective in measuring the quality of the model's output. The target label was manually labeled and was not precise on a pixel level. For example, the hand-drawn label does not adapt to nerve fibers with different widths, so pixels on the edge of a thick nerve fiber might not be labeled. The effect of a mislabeled pixel prominently affects the Dice score because of the scarcity of nerve labels. This results in the model's predictions getting poor Dice scores on visually identical nerve labels. The width of nerve fibers is irrelevant for identifying the crossings. Thus, we designed a more forgiving scoring metric based on a Dice score that tolerates predicted pixels to be k pixels off the ground truth. We counted predicted pixels that are k pixels away from any ground truth pixel as TP_k (True Positive within k)and defined a modified Dice score as:

$$Modified\_Dice(k, P, T) = \frac{|TP_k|}{(|P| + |T \cup TP_k|)/2}$$

When k is 0, the normal Dice score is a special case of this measure.

For nerve labeling, we compared the accuracy of three models: A U-Net pre-trained for abnormality segmentation on a dataset of brain MRI volumes [3] fine-tuned on our dataset, a randomly initialized Holistic Edge Detector (HED) (i.e., trained from scratch on our data), and HED pre-trained on BSDS500 and fine-tuned on our data. The hyperparameters for training the networks were determined by grid search as above (learning rate and weight decay). We trained the models for 200 epochs using data augmentation, and evaluated the models' predictions on the test set using the Dice score and the modified Dice score with k set to 1 and 3. All models were trained on two NVIDIA GeForce GTX 1080 Ti's in about an hour.

3.2 Evaluating Dermis Model

For the dermis models, we compared the performance of three versions: the same U-Net model as above, a randomly initialized DeepLabV3 model, and a pre-trained DeepLabV3 fine-tuned on our dataset. Each is similarly tuned and trained using grid search over the hyperparameters. The U-Net model was trained on four NVIDIA GeForce GTX 1080 Ti cards in about half an hour. The DeepLabV3 model trained on a single NVIDIA GeForce GTX 3090 for 1.5 h.

4 Results

We evaluate our model at two levels. For the end-to-end counting pipeline, we use 5-fold cross validation and compare the results of the model with that of a human expert using Pearson correlation. To evaluate the individual components, for the sake of time, we train on a subset of the data for each model and compare their performance on a held-out portion. For this comparison, we use 84 training images and 10 held-out images.

4.1 Nerve Labeling Results

The results of the nerve models on the 10 held-out images are shown in Table 1. Under all evaluation metrics, the HED model pre-trained on BSDS500 dataset outperforms the pre-trained U-Net model and randomly initialized HED model.

Table 1. Nerve labeling scores shown in format mean (standard deviation)

Model	Dice score	Modified dice score k = 1	Modified dice score k = 3
U-Net	0.585 (0.083)	0.814 (0.085)	0.918 (0.079)
HED not pretrained	0.549 (0.083)	0.778 (0.089)	0.885 (0.085)
HED transfer learning	**0.611** (0.067)	**0.845** (0.054)	**0.950** (0.036)

4.2 Dermis Labeling Results

The resulting performance of each dermis model on the 10 held-out images is shown Table 2. The DeepLabV3 model shows a clear performance increase over U-Net, with a slight improvement (numerically) by using transfer learning.

Table 2. Dermis labeling scores shown in format mean (standard deviation)

Model	Dice score
U-Net	0.958 (0.027)
DeepLabV3 not pretrained	0.979 (0.012)
DeepLabV3 transfer learning	**0.986** (0.007)

4.3 Crossing Count Results

The results of the correlation comparison on the whole pipeline on each of the 5 folds are shown in Table 3. Each line indicates a model trained on a subset of the data and its correlation with the training expert on held-out data, so each score represents the model's generalization performance.

Table 3. Correlation between model prediction and ground truth over each fold. p-values below 0.001 are shown as 0.

Fold	Correlation	p
1	0.835	0.00
2	0.465	0.04
3	0.772	0.00
4	0.931	0.00
5	0.760	0.00

The model's counts were correlated with the training expert at an average of 0.753 with standard deviation 0.156 over the 5 folds. To compare the model's performance with a second expert, we obtained another set of counts on the 10 validation images used for the evaluation of the components, and trained the pipeline on the 84 training images. The two experts' counts were correlated with each other at 0.467 ($p = 0.173$), while our model is correlated with the training expert at 0.834 ($p = 0.002$). This demonstrates that the model correlates strongly with the training expert—even more than the second expert, who was trained by the first!

Examples of the model results are shown in Fig. 4. Overall, the model produces quality output that resembles the expert's label and gives a more consistent count than a human expert. Most variations are caused by the model making a different but reasonable judgment of the dermal-epidermal boundary, for example in the second image pair.

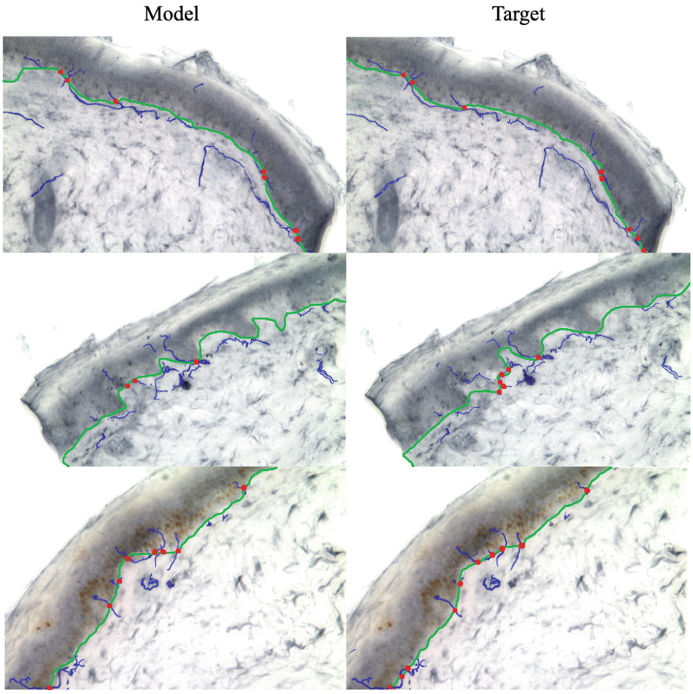


Fig. 4. Comparison between model output (left) and training target labeled by expert (right) on the test set. Blue: nerves; Green: dermal-epidermal boundary; Red: nerve crossings. (Color figure online)

5 Discussion

We proposed a fully automated, end-to-end system for detecting and counting nerve fibers in the skin which cross the dermal-epidermal boundary. We found that the model was highly correlated with the expert's labeling of the data, while a second evaluator was not statistically significantly correlated with the expert. Finally, using augmentation and cross-validation we show that even with very limited training data, the model still has good performance and generalization. A possible direction for future work is to generalize the model to different magnifications and microscope resolutions—this is not automatically handled by the model since varying these factors does not preserve the proportion of physical distance to pixel distance. One approach which does not require any changes to the model is to take a picture of a ruler with the microscope at the desired magnification and resolution to obtain this physical to pixel distance measure, and scale the input image accordingly. Another opportunity for improvement is to generalize to different colors of stains used in the samples, since this could vary between labs. These limitations could be simply solved by obtaining and training on more data, but perhaps other methods such as scale and color augmentations could be investigated.

Acknowledgements. Liam Tan contributed to the nerve crossing detection algorithm, and Maria Rodriguez and Yasmine Oliva-Illera completed manual annotations of the data. Computational support from NSF grants 2120019, 1730158, 1541349 and 2100237.

References

1. Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **33**, 898–916 (2011)
2. Bergwerf, H., Bechakra, M., Smal, I., Jongen, J.L.M., Meijering, E.: Nerve fiber segmentation in bright-field microscopy images of skin biopsies using deep learning. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), pp. 232–215. https://doi.org/10.1109/ISBI.2019.8759504
3. Buda, M., Saha, A., Mazurowski, M.A.: Association of genomic subtypes of lowergrade gliomas with shape features automatically extracted by a deep learning algorithm. Comput. Biol. Med. **109**, 218–225 (2019)
4. Chen, L., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. CoRR, arXiv:abs/1706.05587 (2017)
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996)
6. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: the pascal visual object classes challenge: a retrospective. Int. J. Comput. Vis. **111**, 98–136 (2015)
7. Al-Fahdawi, S., Qahwaji, R., Al-Waisy, A.S., et al.: A fully automatic nerve segmentation and morphometric parameter quantification system for early diagnosis of diabetic neuropathy in corneal images. Comput. Methods Prog. Biomed. **135**, 151–166 (2016)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR; abs/1512.03385 (2015)
9. Niklaus S.: A Reimplementation of HED using PyTorch (2018). https://github.com/sniklaus/pytorch-hed
10. Pal, A., Garain, U., Chandra, A., Chatterjee, R., Senapati, S.: Psoriasis skin biopsy image segmentation using Deep Convolutional Neural Network. Comput. Methods Prog. Biomed. **159**, 59–69 (2018)
11. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
12. Xie, S., Tu, Z.: Holistically-nested edge detection. CoRR ;abs/1504.06375 (2015)
13. Zhang, D., Huang, F., Khansari, M., et al.: Automatic corneal nerve fiber segmentation and geometric biomarker quantification. Euro. Phys. J. Plus **135**, 266 (2020)