A Machine Learning Approach to the Automatic Classification of C. elegans

Poojit Hegde[a]

[a]Stanford University, CS 279

ABSTRACT

In this paper we seek to automate the classification of *C. elegans* using convolutional neural networks. Specifically, we use a fully convolutional U-net with a resnet encoder to perform semantic image segmentation of images containing C. elegans, and the images of the roundworm animals are obtained from Broad Bioimage Benchmark Collection (BBBC). By labeling the images pixel-wise as "alive", "dead", or "background", the specific classification tasks we aim to accomplish are classifying entire images on whether they have been treated with amoxycillin or not, as well as classifying individual roundworms as dead or alive. Our preliminary results from the limited data set demonstrates that our trained model performs extremely well at classifying entire images as treated vs. untreated, and performs well at classifying individual roundworms as well.

Keywords: CNN, C Elegans, Machine Learning, Classifier

1. INTRODUCTION

The nematode Caenorhabditis elegans, abbreviated as C. elegans, is a roundworm that is very useful in researching the effectiveness of various anti-infection drugs and therapies. Enterococcus faecalis is a pathogen known to cause many kinds of health problems such as urinary tract infections. To determine the effectiveness of ampicillin as a drug therapy to treat animals infected with Enterococcus faecalis, a method may be to determine its efficacy in C. elegans.

Microscopy images of C. elegans treated with ampicillin after infection seem to display an "alive" phenotype, with a more curved shape and smooth texture. On the other hand, untreated C. elegan exhibit a "dead" phenotype, with a straight shape and somewhat uneven texture.

There are two main tasks that I aim to achieve. First, given an image containing at least one C. elegans, I aim to identify the image is of C. elegans which have been treated with amoxycillin, or not. The second task is to classify each individual C. elegans in an image as "alive" or "dead" based on its morphological visual appearance.

The WormToolbox through the open-source Cell Profiler application has features for automatic classification of C. elegans. I hope to approach this problem with a separate implementation, with automatic classification via semantic segmentation using a U-net fully convolutional neural network with a resnet encoder.

2. DESIGN AND IMPLEMENTATION

I used the C.elegans infection live/dead image set version 1 provided by Fred Ausubel and available from the Broad Bioimage Benchmark Collection. This dataset contains 100 different images, with 52 being untreated, and 48 being treated with amoxycillin. Each image consisted of numerous C. elegans, approximately 10-20, with the pixel values being in variety, but all grayscale. The image set also included binary versions of the images, in which the C. elegans had pixel values [255,255,255] and the background was [0,0,0]. There was also a third segment of the set that included each individual C. elegans extracted as a binary image. I used all three of these sets in my design, the first being the primary dataset, with the other two being used in the algorithmic implementation. Each image is 696 by 520 by 3 pixels. Although the images are 3-dimensional (with 3 color channel), all the values are the same for a given pixel (such as [2,2,2]), so for purposes of notation for this paper I will refer to them as a single pixel value (such as 2 for [2,2,2]).

The primary parts of the pipeline are as follows. First, training data is prepared. Then, I use the data to train a model that performs semantic image segmentation, or classifies images pixel-wise. Third, I used the trained model to determine the accuracy of the semantic segmentation classifier on a test set of images, for both of the tasks mentioned in the introduction, of classifying entire images as "treated" or "untreated", and determining if each C. elegans in an image is either "dead" or "alive".

2.1 Preparing the Training (and Validation/Test) Data

For preparation of the training data, I divided the 100 image dataset into two parts with a 4:1 ratio, with 80 images being used for training and the other 20 being used for the validation / test set. The separation is chosen randomly. The prepared data is divided into four folders: input training images, input validation images, output training images, and output validation images. The input training images and input validation images are obtained by simply resizing the corresponding original images for it to be compatible with the U-net architecture, with each dimension being a multiple of 32 pixels. Hence the images to be trained are all resized to 704 pixels by 512 pixels.

The annotated images (the expected output of the model), require labeling each pixel of the image as belonging to a dead C. elegans, belonging to an alive C. elegans, or being part of the background. The image set provided a set of each extracted individual C. elegans, but they were not labeled. Thus, I manually labeled every one of the 1500+ C. elegans as each being alive (pixel value 1) or dead (pixel value 2). I stored these manual labels in a text file for use in my preparation algorithm. I used a loosely defined visual heuristic of "curvature" to determine the status. If the C. elegans was relatively straight, I labeled it as "dead", and if it was not, and had at least some noticeable curvature, then I labeled it as "alive".

Then, I used computer vision to make sure that the annotated images were appropriately labeled with each pixel belonging to an "alive" C. elegans being labeled 1, each pixel belonging to a "dead" C. elegans being labeled 2, and each background pixel being labeled as 0. A critical point to note here is that some of the C. elegans overlapped, with some pixels belonging to multiple roundworms. Here, those pixels are labeled with the value corresponding to the latest

individual C. elegans that is labeled. The annotated images are also split 4:1 into training and validation/testing. They are also resized from 690 by 520 pixels to 704 by 512 pixels.

2.2 Training the Model

For training the semantic segmentation model, I used a U-net architecture with a resnet encoder. A U-net is a kind of fully convolutional neural network, and the resnet encoder has been used in recent literature in ML to have good performance for semantic segmentation and classification of C. elegans.

Some of the parameters I set are making both the input and output images 704 by 512 pixels, and the number of classes to be 3 (pixels of 0,1,2 in the output images). The number of epochs for training I used is 10, with a batch size of 2, with 2048 steps per epoch.

Because of the significant computational resources needed for training this model with this architecture, I used the Sherlock computing cluster. I prepared the environment for training on a Sherlock computing node, and wrote a bash script for the submitting the job using GPU resources.

2.3 Determining the Performance

After training model, then comes determining its performance for our two main tasks of classification beyond the validation/test accuracy of the pixel-wise semantic image segmentation at its raw output.

For each of the images in the test set, I use the model on the images to output a segmented image, with 0 for the background pixels, 1 for the predicted pixels belonging to a live C. elegans, and 2 for the predicted pixels belonging to a dead C. elegans.

For predicting the image type, I count the number of pixels in an the predicted image from the model that are 1 or 2, and whichever is higher, that is the decision that is chosen. If there are more pixels with the value 1 than the value 2, then the image is labeled as "untreated". If there are more pixels with the value 2 than the value 1, then the. image is labeled as "treated". Then, for determining the accuracy of the predictions, the predictions are cross-referenced with the actual condition of the image (treated or untreated). The percentage accuracy is determined by the number of images correctly classified over the total number of images in the test set.

For checking the accuracy of the model for determining if individual C. elegans in the image are dead or alive, I compare the predicted image to each of the individual binary images of the C. Elegans from the third component of the original dataset. For every C. elegan image from the binary image dataset of individual roundworms, I scale the image to the same size, and then I check to see what the most common pixel value in that region for the predicted image is. If it is 1, then that C. elegans is labeled as alive, and if it is 2, then that C. elegans is labeled as dead. Then, I compare from the manually labeled truefile.txt document I created with the actual labels for each C. elegans, to determine the percentage that is correct.

2.4 Challenges and Considerations

This design and implementation does have some drawbacks, and some challenges were involved. A major challenge I encountered is dealing with overlapping C. elegans. I initially considered labeling pixels that belong to multiple C. elegans as a separate category for "overlap pixels", but decided against it. Right now, the decision to label overlapped pixels as belonging to one of the C. elegans relatively arbitrarily is not ideal, but I think it suffices, and the accuracy does not noticeably suffer. Comparatively, if I were to label it as an artificial category of "overlap", the issue of what to count that in determining accuracy later on seemed to present more of an issue nevertheless.

Another challenge was the requirement for the input image into the U-net to have dimensions that are multiples of 32 pixels. The images from the dataset are slighlty away from such a number, so I had to decide how to adjust for that, whether to crop images for the dataset, which preseves aspect ratio, or to slightly sacrifice the preservation of aspect ratio to keep the entirety of the view in the training set. I decided to go with the latter, because since altering the aspect ratio, with the interpolation method I used, should not have changed the curvature of the C. elegans significantly if at all, which defines the C. elegans status as "dead" or "alive". If the morphology that we were detecting was transformed by editing the aspect ratio slightly, then I would probably have chosen the former design decision or made a slightly different algorithmic design to accommodate for that.

3. RESULTS

The training loss and accuracy curves over the 10 epochs are shown in figures 1 and 2 respectively. The spike in training loss and drop in training accuracy in epoch 8 is peculiar, and is briefly discussed in the next discussion section.

For our first task of determining if an image is one of C. elegans treated by amycillin, the accuracy of our model in predicting that is 100%, that is, for every one of the 20 test images, it accurately predicted if it was treated or untreated. Figure 4 shows the confusion matrix for task 1.

For the second, more difficult task, of determining the status of each individual C. elegans extracted from the image, the performance is still high, with correct classification of 255 out of the 275 C. elegans. Figure 3 shows the confusion matrix for task 2. 94.0% of the cells that were actually alive were labeled as alive, and 6.0% were falsely labeled as dead. In addition, 92.4% of the cells that were actually dead were correctly labeled as dead, with 7.6% being falsely labeled as alive. This amounts to a total accuracy for the 275 C. elegans as being over 93%.

4. DISCUSSION

The very good performance of the entire image classification task as well as the relatively good performance of the individual C. elegans classification task were very encouraging preliminary results for this project.

However, the relatively small size of the dataset (100 images) limit measurement of robustness of the model. In addition, I used the same randomly selected 20 images for the test and validation

sets, so that is another consideration. Also, some of the parameters/hyperparameters were surely not optimally chosen. This probably explains the increase of training loss and decrease of training accuracy at the 8th epoch. A high learning rate can cause this due to moving away from the local minima and increasing loss.

4.1 Future Work

Future work for this project includes finetuning some parameters. In addition, I could use GANs to try to expand the dataset to see if there can be any meaningful conclusions drawn from that. I also can use data augmentation to artifically expand the dataset in that way.

I also plan to do more research to compare this to state-of-the-art methods, and see if such classification of C. elegans has been done in other ways in the past. I also want to see if some different algorithmic methods and tasks can be performed via semantic segmentation or otherwise with this image set.

This is a high-throughput experiment, but the work so far is still preliminary, and the methodologies I used have potential for improvement.

ACKNOWLEDGMENTS

I'd like to acknowledge the CS 279 teaching team and material for guiding me through learning more about computational biology and image analysis. I'd also like to acknowledge my mentor for the research I do at Stanford Genome Technology Center (SGTC), Alex Kashi, who helped teach me about basic machine learning, and showed me some of the relevant literature. Working on a research project at SGTC over the past few months on classification of red blood cells for ME/CFS (chronic fatigue syndrome) patients was my first introduction to machine learning projects. This helped inspire me to use U-nets would work for this semantic segmentation task. I'd also like to acknowledge Lastly, I'd like to acknowledge my dog for being my best friend.

REFERENCES

[1] Ljosa, V., Sokolnicki, K. & Carpenter, A. Annotated high-throughput microscopy image sets for validation. Nat Methods 9, 637 (2012). https://doi.org/10.1038/nmeth.2083

[2] Wählby, C., Kamentsky, L., Liu, Z. et al. An image analysis toolbox for high-throughput C. elegans assays. Nat Methods 9, 714–716 (2012). https://doi-org.stanford.idm.oclc.org/10.1038/nmeth.1984

[3] Silverman, G., Luke, C., Bhatia, S. et al. Modeling Molecular and Cellular Aspects of Human Disease Using the Nematode Caenorhabditis elegans. Pediatr Res 65, 10–18 (2009). https://doi-org.stanford.idm.oclc.org/10.1203/PDR.0b013e31819009b0

[4] Durant TJS, Olson EM, Schulz WL, Torres R. Very Deep Convolutional Neural Networks for Morphologic Classification of Erythrocytes. Clin Chem. 2017 Dec;63(12):1847-1855. doi: 10.1373/clinchem.2017.276345. Epub 2017 Sep 6. PMID: 28877918.

[5] Mittal, Aditi. "Introduction to U-Net and Res-Net for Image Segmentation." Medium, Towards Data Science, 10 June 2019, towardsdatascience.com/introduction-to-u-net-and-res-net-for-image-segmentation-9afcb432ee2f.

[6] Ronneberger, Olaf, et al. "U-Net: Convolutional Networks for Biomedical Image Segmentation." Lecture Notes in Computer Science Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, 2015, pp. 234–241., doi:10.1007/978-3-319-24574-4_28.

5. FIGURES

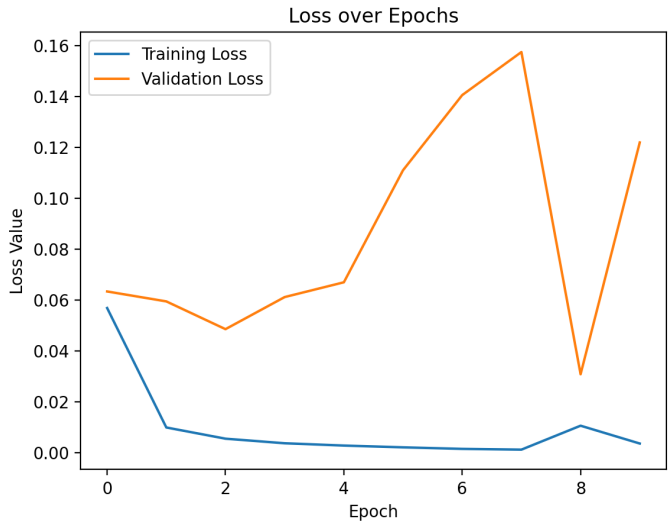

Figure 1. This figure depicts the loss for training and validation over the 10 epochs (pixel-wise for the semantic segmentation).

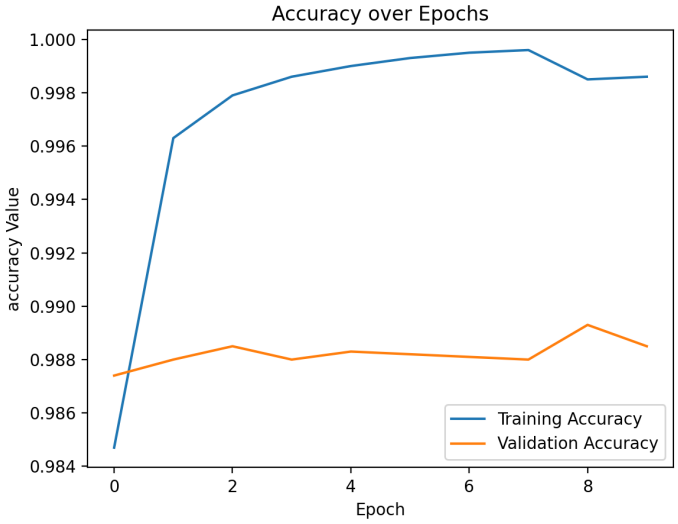

Figure 2. This figure depicts the accuracy for training and validation over the 10 epochs (pixel-wise for the semantic segmentation).

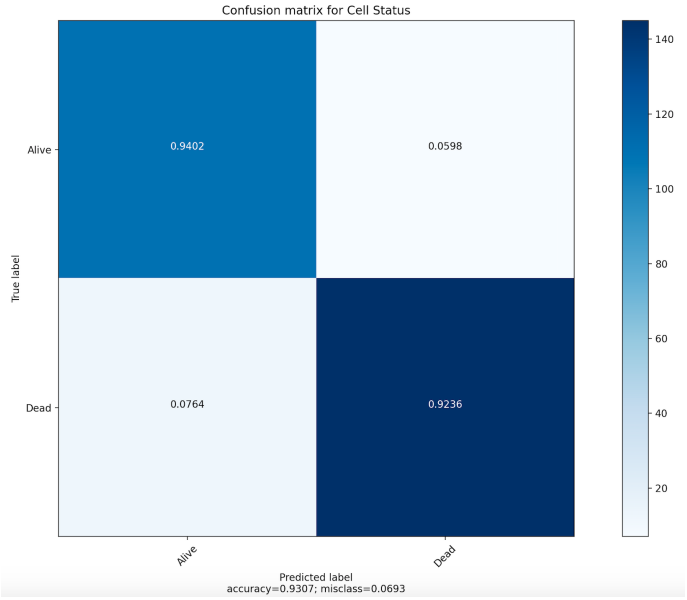

Figure 3. This is a confusion matrix showing the accuracy of prediction between entire images being treated or untreated.

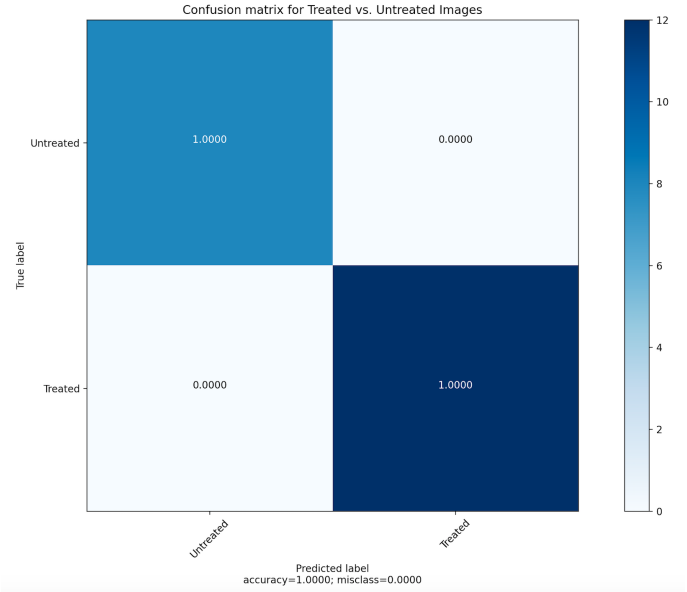

Figure 4. This is a confusion matrix showing the accuracy of prediction between individual animals in images being dead or alive.