Title

Quantification and Accuracy Evaluation of Tau Tangle Distribution in Postmortem Brain Microscopy Images from Patients with Alzheimer's Disease Using U-Net Object Segmentation Model

Author

Bennecke, Andrew Richard

Publication Date

2023

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Quantification and Accuracy Evaluation of Tau Tangle Distribution in Postmortem Brain Microscopy Images from Patients with Alzheimer's Disease Using U-Net Object Segmentation Model

A thesis submitted in partial satisfaction of the requirements for the degree Master of Science in Bioinformatics

by

Andrew R. Bennecke

2023

ABSTRACT OF THE THESIS

Quantification and Accuracy Evaluation of Tau Tangle Distribution in Postmortem Brain Microscopy Images from Patients with Alzheimer's Disease Using U-Net Object Segmentation Model

by

Andrew R. Bennecke

Master of Science in Bioinformatics

University of California, Los Angeles, 2023

Professor Daniel Jacob Tward, Chair

Alzheimer’s Disease is a progressive and fatal neurodegenerative disease which affects millions of people around the world. The pathophysiology of the disease is characterized by the accumulation of neuritic amyloid plaques and neurofibrillary tau tangles within the hippocampus and many surrounding structures. Tau tangles, in particular, are commonly used to identify the stage of disease progression. Currently, only the presence or absence of tau tangles, together with simple staging information, in specific brain regions is noted at autopsy. An improvement to this approach is to measure the distribution of tau tangles across large brain samples in order to better characterize the progression of the disease. In this work, we build a framework for comparing the ability of different machine learning models to identify the locations of tau tangles in postmortem neural microscopy images. In particular, we focus on the development of a set of

software tools which transform probability heatmaps into a set of region proposals for all the tau tangles within an image. We then construct two different machine learning models and compare their performance using a precision-recall (PR) framework.

The thesis of Andrew R. Bennecke is approved.

Pavak Kirit Shah

Shantanu Hemachandra Joshi

Daniel Jacob Tward, Committee Chair

University of California, Los Angeles

2023

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Original word(s)	Abbreviation
Alzheimer's Disease	AD
Amyloid/Tau(Neurodegeneration)	A/T(N)
Average Precision	AP
False Negative	FN
False Positive	FP
Intersection Over Union	IOU
Magnetic Resonance Imaging	MRI
Mean Average Precision	mAP
Non-Maximum Suppression	NMS
Precision-Recall	PR
Region-based Convolutional Neural Network	R-CNN
Tagg Image File	tif
True Negative	TN
True Positive	TP

1: Introduction

Alzheimer's disease (AD) is a progressive and ultimately fatal neurodegenerative disease that affects over 6 million Americans. Economically, the disease is estimated to cost the nation $345 billion per year, which is expected to rise to $1 trillion by 2050 as the population ages [https://www.alz.org/media/Documents/alzheimers-facts-and-figures.pdf, accessed 03/2023]. While it is known that 1 in 3 seniors dies with AD or another form of dementia, the disease is currently only diagnosed at autopsy from the presence of amyloid plaques and tau tangles in brain tissue [1,2].

There are several biomarkers of AD that are being developed for research purposes, which can increase or decrease the certainty that symptoms of dementia in living people are due to the Alzheimer's pathophysiological process. These may contribute to early diagnosis, and have helped to define early stages of the disease such as Mild Cognitive Impairment [3] or Preclinical AD [4]. These biomarkers have now been standardized into the Amyloid / Tau (Neurodegeneration) (A/T(N)) framework [5]. In this standard notation, neurodegeneration is written in parentheses because it currently shows a lack of specificity. However, jointly studying neurodegeneration with other biomarkers may increase this specificity. In particular, tau tangles (unlike amyloid) are known to accumulate in a stereotypical pattern and are useful for disease staging (Braak staging)[6].

A promising approach to linking tau tangles with neurodegeneration is to study their distribution across the whole brain, rather than simply noting presence or absence in key areas which is typical at autopsy. This requires a machine learning approach to identify tangles in microscopy

images, and a geometric approach to reconstruct 3D distributions. Several methods have been developed to attempt this [7][8], but there is no agreed upon framework to compare them or evaluate their accuracy. This is particularly challenging because tau tangle detectors may have very different kinds of outputs, either region proposals (common in natural images) which take the form of bounding boxes with confidences, or probability masks (more common in biomedical images) which take the form of heatmaps with values in the interval [0,1].

In this work, we address this challenge by building a framework for comparing different machine learning techniques that detect tau tangles. We focus on developing a common analysis tool that can transform probability masks into region proposals and evaluate their quality using a precision-recall (PR) framework that has become standard in the analysis of natural images. Moreover, we implement two different techniques based on modern and classical computer vision algorithms, and compare their performance within this PR framework. The primary contribution of this thesis is the development of software tools for accomplishing these tasks; however, further work will be required to make quantitative statements about the disease itself.

2: Methods / Results

2.1: Dataset

Our data was acquired as described in [8], and includes microscopy images of medial temporal lobe tissue sliced every 1 mm and imaged with approximately 2 micron resolution. The microscopy images are stored in the Tagg Image File (tif) format and the image annotations are stored in the NIfTI file format (a standard developed for brain images). Each microscopy image has approximately 8,000 to 14,000 pixels along each axis and has 3 channels (red, green, blue). The annotation files are the same dimensions as the microscopy images and each contains 5-12 randomly selected chunks which were annotated by a neuroanatomist. Within each chunk, every pixel is marked as either background (with the value 1) or as a tau tangle (with the value 2). Pixels outside of annotated chunks are given the value 0. Twelve image-annotation pairs are used to generate the inputs to the models discussed in Section 2.4. One example annotated image is shown in Figure 1, shown at low and high magnification. Along with the microscopy images and associated annotations, the dataset also contains 3D MRIs, which will be used in future work to correlate tau tangle density with neurodegeneration patterns.

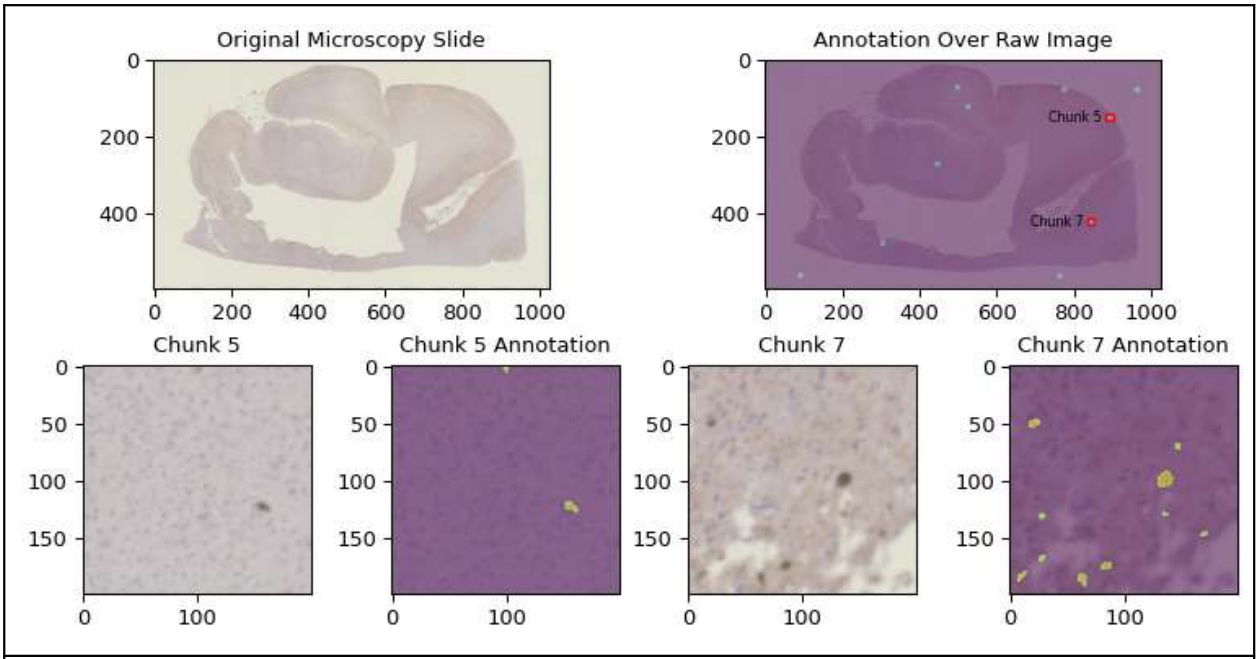

Figure 1: Annotated tau histology image. Top left shows a tau immunostained section of a human medial temporal lobe. Top right shows randomly selected chunks that were completely annotated. Two example chunks, and their annotations are shown on the bottom row, where yellow refers to a tau tangle, and purple refers to anything else.

2.2: Preprocessing

Several images needed to be stitched together due to the fact that the imaging technology used to annotate this data (Seg3D, https://www.sci.utah.edu/cibc-software/seg3d.html) was not able to interactively process files of this size (several GB per file) without significant delay. In order to transform these annotations into viable targets, each annotation chunk and the corresponding image chunk needed to be extracted and stored locally. In this work, a "target" is defined as the annotation of an image, which we would like our machine learning models to predict as accurately as possible.

To accomplish this, each annotation image was parsed through, pixel-by-pixel, until an anchor pixel was identified. In this work, an “anchor pixel” is defined as one with a value of 1 or 2 (indicating this pixel was annotated), and whose upper and left neighbors have a value of 0 (indicating this pixel was not annotated). An anchor pixel corresponds to the upper-leftmost pixel in a 200x200 annotated chunk. Once an anchor pixel was identified, the 200x200 chunk with the anchor pixel as the upper left corner was extracted from the larger annotation image and the corresponding 200x200 chunk in the microscopy image was also extracted. These two chunks were then saved locally as *.tif* files. Then, we continued to parse through the annotation image and extract all remaining annotated chunks. This was repeated for every annotation image. From the 12 annotation files, a total of 155 chunks were extracted, each 200 x 200 pixels. 49 of these chunks contain masks that include the presence of tau tangle and the remaining 106 chunks contain only background. Background was included so that the frequency of tau positive pixels in our annotated set was roughly equal to the frequency of tau positive pixels in our large images.

2.3: Dataset Class

In order to use the pre-built functions and model architectures from the *torchvision* package, a custom dataset class was implemented. To be initialized, this class required a directory containing all of the image chunks, a directory containing all of the corresponding annotation chunks, an optional argument for returning either boolean masks or bounding boxes, and an optional argument for using either the entire dataset or a predetermined subset. An example of a predetermined set would be the training data resulting from an 80/20 train/test split. This dataset was used in *torchvision*’s *DataLoader* class as an input to the machine learning pipeline. When the *DataLoader* requested an item from the dataset, a combination of random

operations were performed on the image and associated annotation. There was a 50% chance that the instance would be flipped horizontally, a 50% chance that the instance would be flipped vertically, and an equal chance that the instance would be rotated either 0^{o}, 90^{o}, 180^{o}, or 270^{o}. The purpose of this was to prevent overfitting and enforce rotation, translation, and reflection invariance by augmenting the dataset with label preserving transformations[9], known as data augmentation. Then, a random 132 x 132 chunk was extracted from the instance for the remaining transformations. Figure 2 shows an example set of operations performed starting with the original image-annotation chunks and ending with the model input. Following these operations, a python dictionary was constructed corresponding to the target and related metadata. This dictionary contained the bounding boxes and masks associated with the annotation, labels of objects in the annotation, unique image identification number, and the area of each bounding box. Ultimately, a tuple containing the randomly transformed image and corresponding target dictionary was returned as an input to the training procedure for the current model.

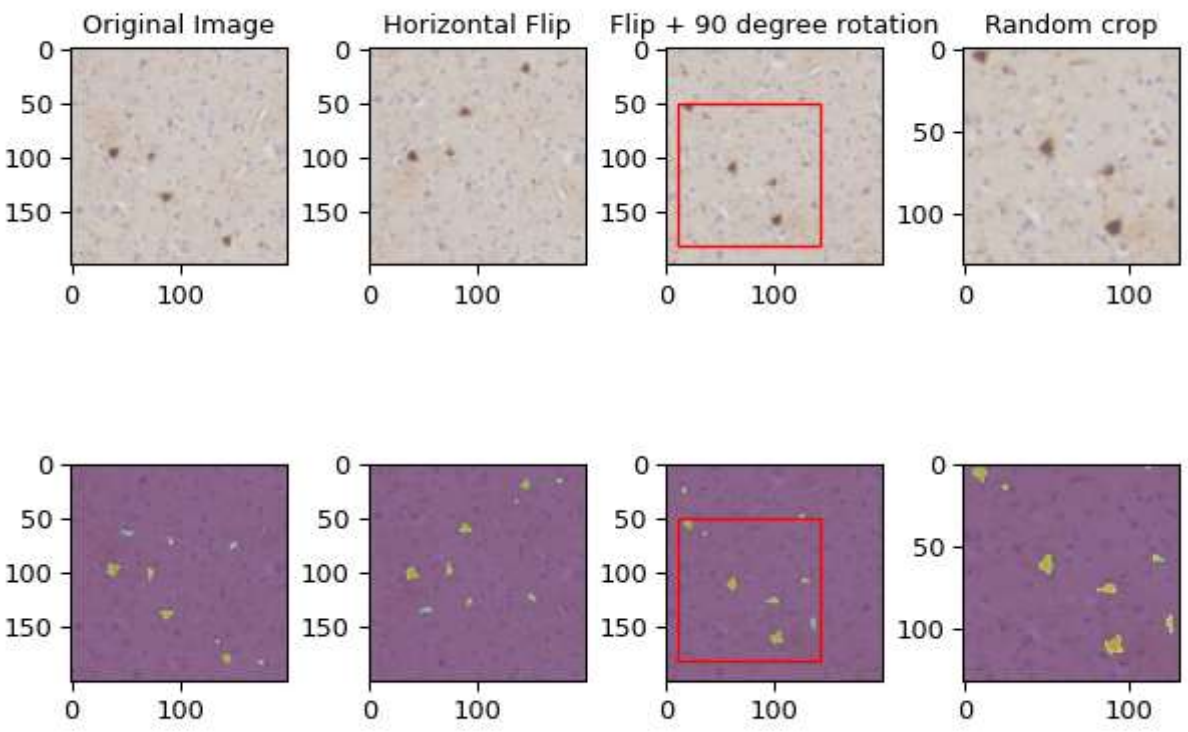

Figure 2: Illustration of data augmentation. Top row shows images, and bottom row shows corresponding annotations. From left to right we see the original image, a flipped image, a rotated image, and a randomly cropped image where the crop is illustrated with a red box in the third column.

2.4: Machine Learning Methods

2.4.1: Model 1 - U-Net model

The U-Net is a type of convolutional neural network proposed by [10]. A differentiating factor of this architecture is that image features are computed iteratively at decreasing resolutions, and heat maps are estimated iteratively at increasing resolutions . Also, when constructing an output probability heatmap, “skip connections” are used to avoid a bottleneck where everything must be computed from low resolution features. In our implementation, an image of size 132x132x3 is passed through the first layer of the network. Several convolution/downsampling layers are applied in sequence to compute features at low resolution, followed by several convolution/upsampling/concatenation layers for defining the output heatmap. The U-Net architecture is illustrated in Figure 3 (left). The script defining the U-Net model can be found at:
https://github.com/twardlab/pathology_detection_andrew/blob/main/unet_arch.py

2.4.2: Model 2 - Linear model

As a baseline for comparison, we include a simpler classification model consisting of a single affine (linear plus addition) transformation. This method is essentially equivalent to linear logistic regression, but we use exactly the same training procedure as for the U-Net. This model is written using the *PyTorch* framework as a neural network with a single *linear* layer. The linear model architecture is illustrated in Figure 3 (right). The script defining the linear model can be found at: https://github.com/twardlab/pathology_detection_andrew/blob/main/linear_arch.py

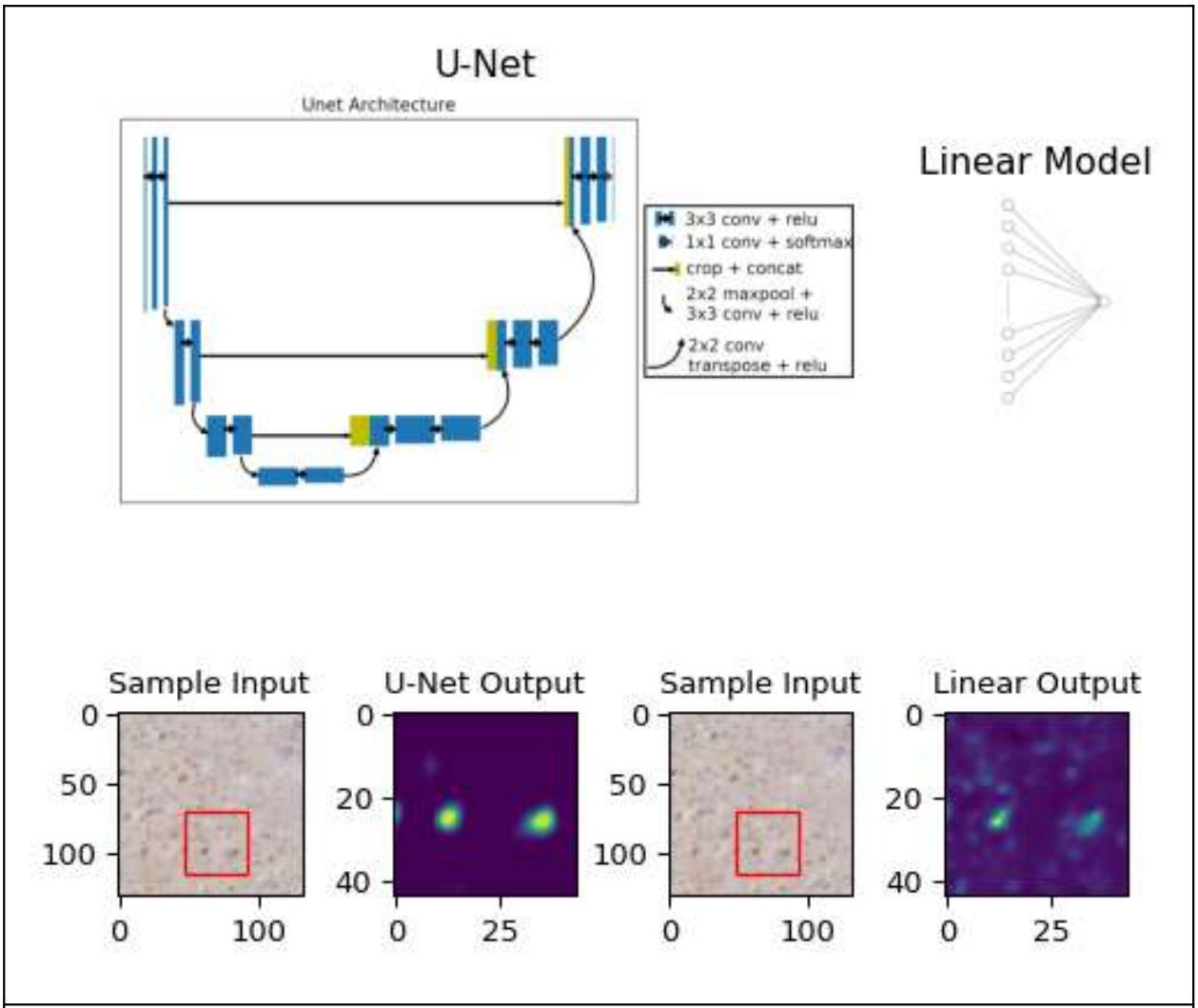

Figure 3: Model architectures and example outputs. The top row shows the architectures for the 2 models used in this work. From left to right, the images on the bottom row show a sample input to the U-Net, the corresponding output from the U-Net, a sample input to the linear model, and the corresponding output from the linear model.

2.4.3: Training

An 80/20 stratified train/test split was performed where image-annotation pairs were grouped by whether or not the annotation only contained background. Due to the imbalance in

annotations with masks, this stratification ensured that every split would contain several instances with tau tangle. Specifically, there were 124 image-annotation pairs used for the training of each fold and the remaining 31 image-annotation pairs were used for testing of each fold. The same splits were used for training and testing of both the U-Net and the linear model. No validation set was used here since we are not considering optimization over any hyperparameters. Cross entropy loss, summed over each pixel, was used as the loss function and the Adam optimizer was used for the optimization step [11] . We use the term “epoch” to denote one pass through the training set during optimization. Training was performed for 500 epochs for each fold. Loss as a function of epoch, for one of our 5 folds, is shown in Figure 4 below. From the figure we can see that the U-Net achieves a lower loss than the linear model, which is expected given the small number of degrees of freedom in the linear model.

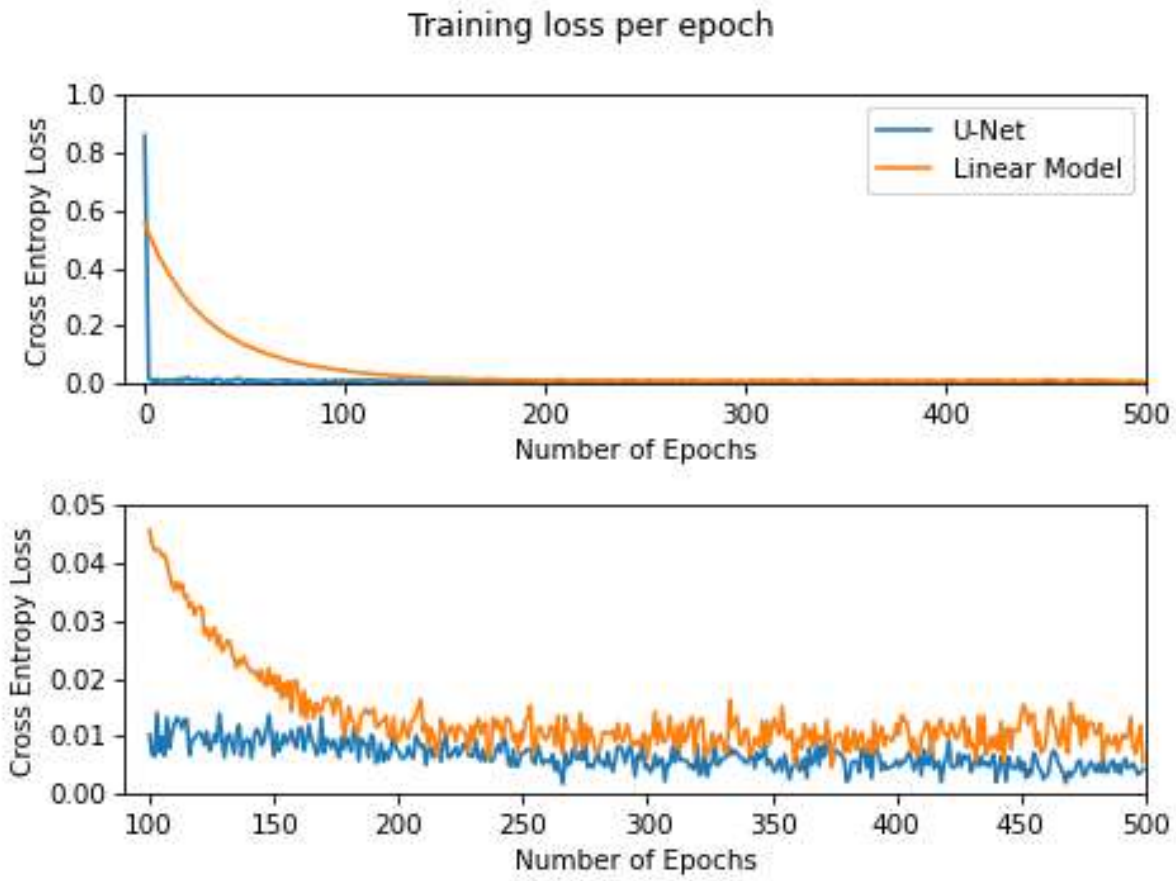

Figure 4: Cross entropy loss as a function of epoch number. For one fold, we show cross entropy loss as a function of epoch number for both the u-net and the linear model. The bottom panel shows the same data zoomed in, by leaving out the first 100 epochs.

2.5 Post-processing

As part of testing the model, each input is transformed into a heatmap, where each pixel represents the probability that a tau tangle is present and ranges from 0 to 1. To remove isolated single pixels that are due to noise, this probability heatmap is then blurred using a Gaussian filter with standard deviation of 1.5 pixels. Then, a set of binarized images was generated from this heatmap by thresholding over every value from 0.01 to 1 using a step size of 0.01. This procedure allowed us to output a set of bounding boxes without having to choose a single optimal threshold. Figure 5 is an example of the set of generated binarized images which arise from this repeated thresholding. The title of each subplot corresponds to the probability threshold used for binarization and the subplot shows the result of thresholding at this value. All pixels with a value greater than the threshold become 1 (yellow) and all pixels with a value less than or equal to the threshold become 0 (purple). The connected components of each binarized image are then defined and bounding boxes are generated surrounding each distinct component. A bounding box is a 4-tuple describing the maximum extent of the component in the x and y direction, in the form: (xmin,ymin,xmax,ymax). Then, a confidence value is assigned to each bounding box. In this work, confidence is defined as the 99th percentile of all the probability values within the bounding box on the original probability heatmap, but more sophisticated approaches could be investigated in the future. This confidence value is needed for the precision-recall analysis described below. Every bounding box from each of these binarized images is added to a set, which is then filtered. This set of bounding boxes contains many bounding boxes corresponding to the same tau tangle. To address this redundancy, bounding boxes were filtered out of this set if they satisfied one of several simple criteria: area of less than 4 pixels, area greater than 400 pixels, or there exists an identical (duplicate) bounding box.

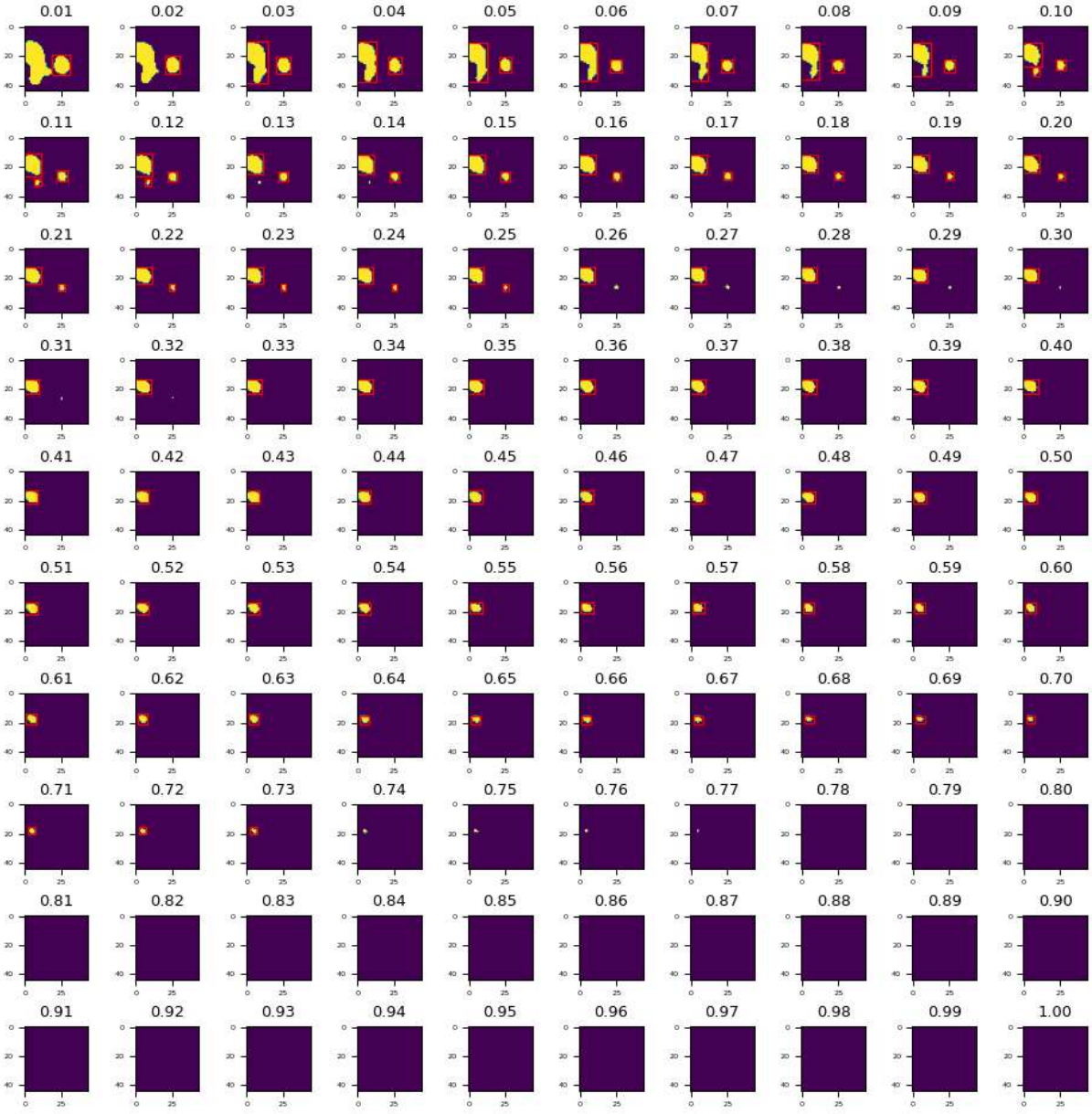

Figure 5: Example of bounding box probability thresholding. Generation and extraction of bounding boxes at all probability thresholds. Notice that the large component at the top left has no bounding box because its area is greater than 400 pixels, and the smaller components toward the bottom right have no bounding box because their area is less than 4 pixels.

After filtering, non-maximum suppression (NMS) (as described in[12]) was performed to choose one bounding box out of each group of overlapping boxes. To accomplish this, first bounding boxes are sorted by their confidence from greatest to least and stored in an input list, and an output list was initialized as empty. Second, the bounding box with highest confidence from the input list was appended to the output list of bounding boxes, and removed from the input list. Third, all bounding boxes in the input list, with an Intersection Over Union (IOU) of greater than 0.5 with respect to this high-confidence bounding box were removed from the input list. Fourth, this process was repeated until no bounding boxes remained in the input list. The value of 0.5 was chosen heuristically, but could be potentially learned from data as stated in [12]. A summary of our post processing steps for one image is shown in Figure 6. We notice that there are still multiple bounding boxes for the same component, and improving the NMS step will be the subject of future work.

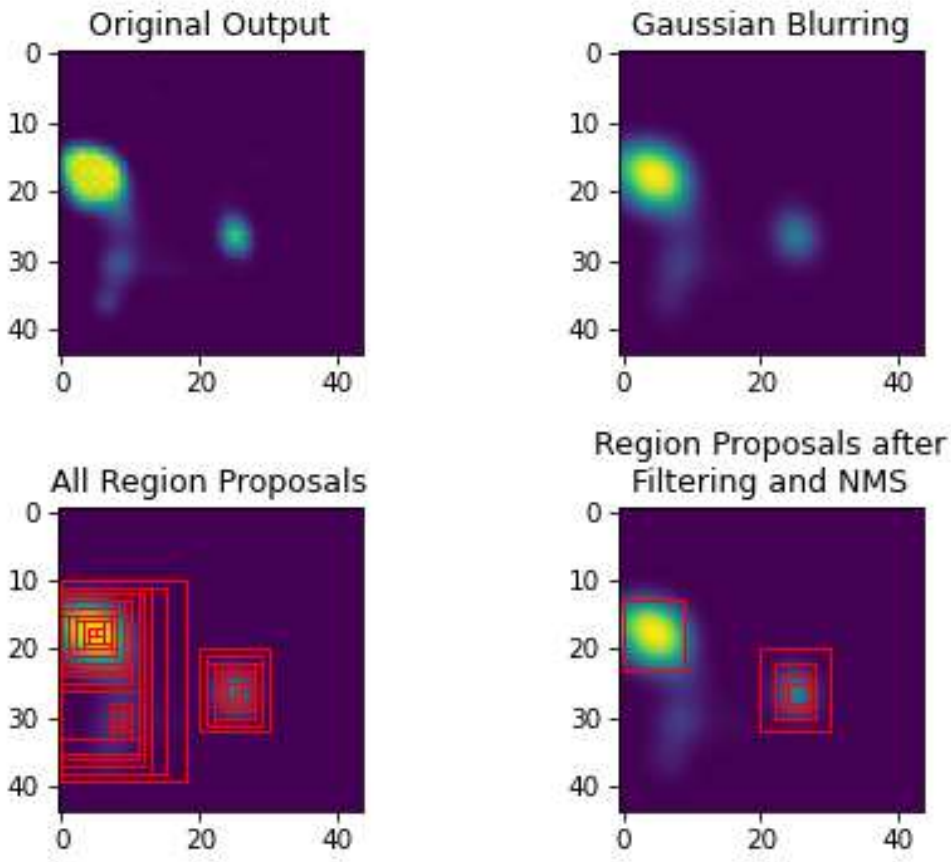

Figure 6: Example of post processing steps. The upper left figure shows the original prediction from the model. The upper right figure shows the original prediction after undergoing a Gaussian blur. In the lower left figure, each red box corresponds to a region proposal generated from the methods discussed in the above section. The lower right figure shows the remaining region proposals after filtering and performing NMS.

2.6 Performance Quantification

Several metrics are used in quantifying the performance of these models. IOU is computed as the area of the intersection between two bounding boxes divided by the area of the union between two bounding boxes. Precision is computed as the number of True Positives (TPs) divided by the sum of the total TPs and False Positives (FPs). Recall is computed as the number of TPs divided by the sum of the total TPs and False Negatives (FNs). For a single IOU threshold, a set of (precision, recall) pairs, parameterized by confidence, are used to construct a PR curve. Average precision (AP) is the area under this PR curve. The final metric used to define model performance is mean average precision (mAP), which is the mean of the set of average precisions from each IOU threshold. We consider all IOU thresholds from 0.5 to 0.95 in steps of 0.05. These steps are illustrated as an algorithm in Figure 7. This PR analysis allows us to compare models without choosing specific thresholds, and is independent of True Negatives (TN) which cannot be meaningfully defined in these cases and are necessary for a more standard receiver operating characteristic analysis.

```
Initialize empty set for all PR curves
for all IOU thresholds (i) in 0.5 : 0.05 : 1 do
    Initialize empty sets for precision and recall
    for all confidence thresholds (c) in 0.01 : 0.01 : 1 do
        Initialize counts for TP, FP, FN to 0
        for all results in model output do
            prob ← Probability heatmap
            ann ← Corresponding ground truth annotation
            Construct set of ground truth bboxes from ann
            Initialize empty set of prediction bboxes
            for all probability thresholds (p) in 0.01 : 0.01 : 1 do
                Construct set of prediction bboxes at p
                Compute confidence for each bbox and append to list of
                 all prediction bboxes
            end
            Remove duplicate bboxes and those with an area too small or
             too large
            Perform Non-Maximum Suppression on list of prediction
             bboxes
            Compute number of TP, FP, and FN for the ground truth
             and prediction sets and add to total TP, FP, FN counts
        end
        Compute precision and recall, and append to respective sets
    end
    Append (precision, recall) curve to set of all PR curves
end
```

Figure 7: Algorithm for generation of PR curves.

In Figure 8, we show PR curves comparing the U-Net model to the linear model, for a single IOU threshold and a single fold (the same fold for both models). Notice that the linear model never outputs a confidence greater than 0.01, and so only outputs false negatives. This is consistent with our expectation that a linear model will be insufficient to classify tau tangles, given variations in foreground and background image intensity throughout the brain images.

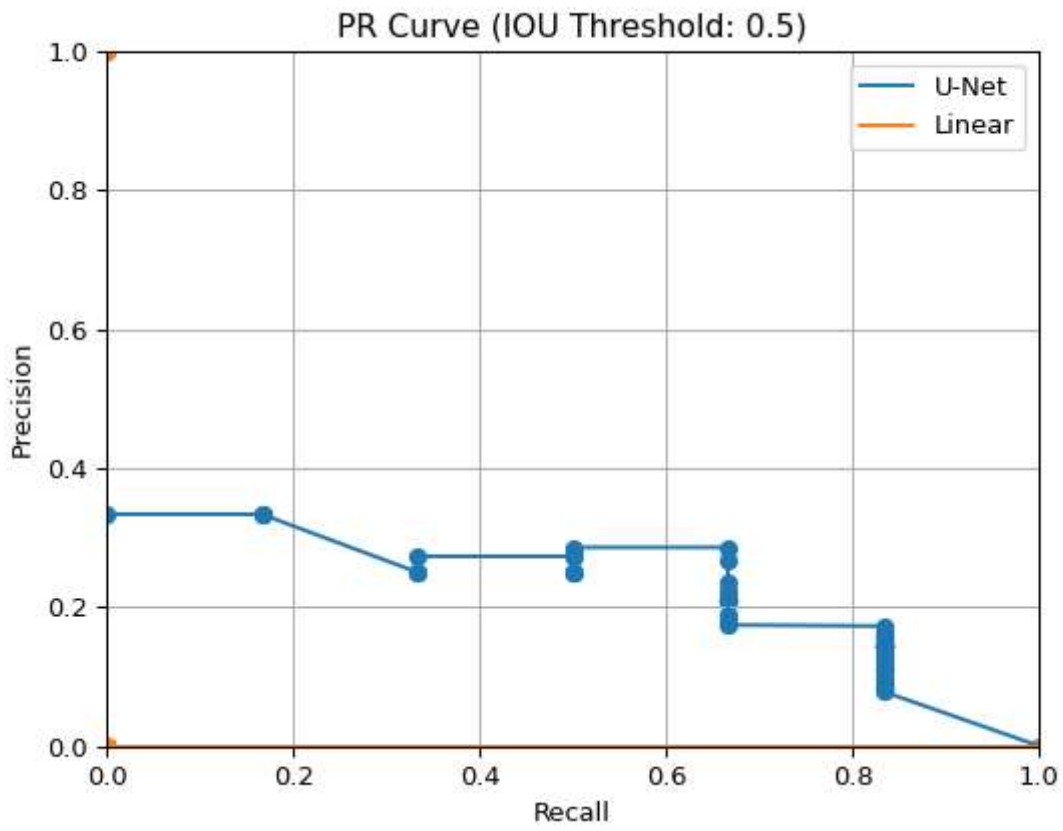

Figure 8: Comparison of the two methods using PR curves. This figure shows the PR curves generated for the U-Net and Linear models over the same fold at a fixed IOU threshold of 0.5.

In Figure 9, we show mAP for each of our 5 folds. We notice that the linear model leads to a value of 0 in all cases as described above. The U-Net performs better over every fold, but its mAP is still quite low given the large number of false positives resulting from our simple NMS procedure.

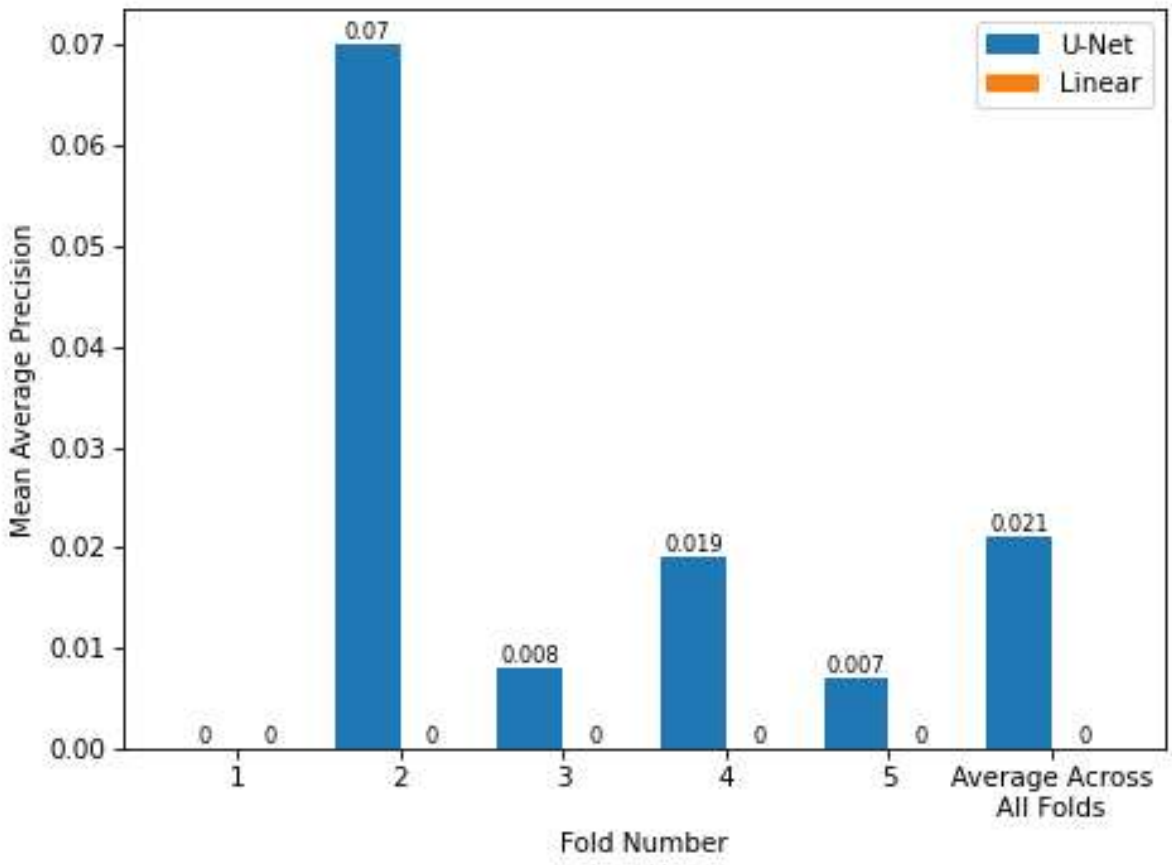

Figure 9: Comparison of final mean average precisions. The first 5 groups compare the mAP between the U-Net and Linear models at each fold. The rightmost group compares the average mAP across all 5 folds.

3: Discussion

First, we built a set of software tools for constructing bounding boxes and assigning confidence to those bounding boxes from a U-Net output. The purpose of these tools is to evaluate the quality of outputs using the PR framework, which is standard for object segmentation tasks involving natural images, but is not typical for object segmentation tasks involving biomedical images. Second, we trained two models for detecting tau tangles in microscopy images from the medial temporal lobe using and compared their performance using this set of software tools. As expected, we found that the linear model was insufficient for detecting tau tangles.

There were various limitations in filling the knowledge gap. First, the NMS/filtering step must be improved to eliminate all of the incorrect bounding box predictions generated from the probability thresholding portion of the algorithm. The presence of these additional predictions leads to an increase in the total number of false positives, which causes a decrease in precision at every IOU threshold. This decrease in precision at each IOU threshold contributes to an overall decrease in mAP. Therefore, increasing the ability of the NMS/filtering step to eliminate these redundancies would lead to an increase in the overall quality of the model's predictions. Second, the method for assigning confidence scores to each bounding box must be improved. The method used in this work was rather crude. A more sophisticated method for generating a confidence score would likely have a stronger mathematical foundation and likely lead to better predictions. Finally, the small quantity of data used in training and testing likely prevented the model from making more accurate predictions with higher confidence. Therefore, a larger amount of data for training and testing would likely lead to an increase in the amount of true positives and a

decrease in the number of false positives. These two changes would contribute to an increase in precision, recall, and ultimately, mAP.

There were also various limitations in the generalizability of our set of software tools. In this work, we developed a procedure for evaluating the performance of machine learning models which output a heatmap. Another important class of machine learning models to which we would like to compare our results are those which directly output a set of bounding boxes, such as those in the region-based convolutional neural network (R-CNN) family. While we designed our method to be compatible with these types of outputs, we have not yet explicitly included them in our comparisons. This type of output, as opposed to heatmaps, is essential for generating counts of tau tangles relative to counts of healthy neurons, which is a biologically meaningful measure of disease severity. For example, in the work of [7], counts of tau tangles were not estimated. Instead, a measure of "tau burden" was computed directly from the probability heatmaps.

Along with addressing the aforementioned limitations, future research could extend this work in several ways. One way that this work could be extended is by developing algorithms that detect all the tau tangles within the dataset and map out their densities in a 3-dimensional space. The distribution of these 3D structures could be correlated with other AD biomarkers, such as amyloid plaque deposits, and other measures of neurodegeneration determined from MRI. The tools that we have built will help when deciding on the best machine learning models for this task.

References

1. Jack CR, Albert MS, Knopman DS, McKhann GM, Sperling RA, Carrillo MC, et al. Introduction to the recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease. Alzheimers Dement. 2011;7: 257–262.

2. McKhann GM, Knopman DS, Chertkow H, Hyman BT, Jack CR Jr, Kawas CH, et al. The diagnosis of dementia due to Alzheimer's disease: Recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease. Alzheimers Dement. 2011;7: 263–269.

3. Albert MS, DeKosky ST, Dickson D, Dubois B, Feldman HH, Fox NC, et al. The diagnosis of mild cognitive impairment due to Alzheimer's disease: recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease. Alzheimers Dement. 2011;7: 270–279.

4. Sperling RA, Aisen PS, Beckett LA, Bennett DA, Craft S, Fagan AM, et al. Toward defining the preclinical stages of Alzheimer's disease: Recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease. Alzheimers Dement. 2011;7: 280–292.

5. Jack CR Jr, Bennett DA, Blennow K, Carrillo MC, Dunn B, Haeberlein SB, et al. NIA-AA Research Framework: Toward a biological definition of Alzheimer's disease. Alzheimers Dement. 2018;14: 535–562.

6. Braak H, Braak E. Staging of Alzheimer's disease-related neurofibrillary changes. Neurobiol Aging. 1995;16: 271–8; discussion 278–84.

7. Yushkevich PA, Muñoz López M, Iñiguez de Onzoño Martin MM, Ittyerah R, Lim S, Ravikumar S, et al. Three-dimensional mapping of neurofibrillary tangle burden in the human medial temporal lobe. Brain. 2021;144: 2784–2797.

8. Tward D, Brown T, Kageyama Y, Patel J, Hou Z, Mori S, et al. Diffeomorphic Registration With Intensity Transformation and Missing Data: Application to 3D Digital Pathology of Alzheimer's Disease. Front Neurosci. 2020;14: 52.

9. Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. Journal of big data. 2019;6: 1–48.

10. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. Lecture Notes in Computer Science. 2015. pp. 234–241. doi:10.1007/978-3-319-24574-4_28

11. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. arXiv [cs.LG]. 2014. Available: http://arxiv.org/abs/1412.6980

12. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv [cs.CV]. 2013. pp. 580–587. Available: http://openaccess.thecvf.com/content_cvpr_2014/html/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.html