Automated Classification of Iron Accumulation in Patients with Alzheimer's Disease

University of Turku
Department of Computing
Master of Science Thesis
Health Technology
July 2025
Niek Pennings

Supervisors:
Tapio Pahikkala (University of Turku)
Louise van der Weerd (Leids Universitair Medisch Centrum)
Jouke Dijkstra (Leids Universitair Medisch Centrum)

UNIVERSITY OF TURKU
Department of Computing

Niek Pennings: Automated Classification of Iron Accumulation in Patients with Alzheimer's Disease

Master of Science Thesis, 58 p., 4 app. p.
Health Technology
July 2025

Alzheimer's disease (AD) is becoming one of the deadliest diseases for elderly people. Although considerable research is being conducted to gain a deeper understanding of AD, many questions about the pathology of the disease remain unanswered. One of these questions is about the role of iron in the disease. To evaluate the spreading pattern of iron in the brain, researchers histologically examine tissue sections from different brain regions of healthy and diagnosed brain donors post-mortem. Currently, they manually evaluate the extent of iron accumulation, but this process suffers from inter- and intra-variability among human experts and is time-consuming.

In this thesis, a method that includes a segmentation and classification model is proposed to automate the task of labelling histological images of brain regions. The goal of the segmentation model is to extract the grey matter from the image, as only the grey matter is relevant to this research. The result of the segmentation serves as the input for the classification model. The images are pre-processed beforehand, namely, they are cropped, downscaled, and the intensities are scaled. The proposed method uses the nnU-Net framework for the segmentation model. The produced segmentation masks are upscaled, and the grey matter is extracted from the original images. After that, the images are downsampled by a factor of 4, divided into patches, and a feature vector is created from each patch. The resulting feature vectors act as input for the classification model. The Clustering-constrained Attention Multiple Instance Learning (CLAM) framework is used for classification.

Our method yields satisfactory results for the segmentation model, outperforming human annotation, with an average Dice score of approximately 0.86 and an average NSD score of around 0.85. This means that the proposed method of using nnU-Net for the segmentation is usable as the input for the classification model. The classification model is unable to learn from the patches. It is inconclusive whether this is due to the experimental design or a software bug.

Keywords: segmentation, classification, nnu-net, clam, alzheimer's disease, iron accumulation, histopathological, machine learning

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

1.1.1 Epidemiology

Worldwide, Alzheimer's disease (AD) is the main cause of the over-arching condition called dementia [1]. In 2019, the Alzheimer International Organisation estimated that there were 55 million people in the world with dementia. This is predicted to triple in 2050 to around 139 million people. It is rapidly rising as the cause of death for many elderly people. In addition to fatality, AD is also a costly disease due to the disease burden in the years before death. The cost is predicted to double from 1.8 trillion USD in 2019 to 3.8 trillion USD in 2030. [2].

1.1.2 Pathogenesis

At the beginning of the 1950s, it was discovered that iron accumulates in the brains of patients with AD. Back then, it was merely found to be a correlation, and little was known about the pathogenesis[3]. More recent research has found that iron levels in tissue, specifically in the inferior temporal cortex, are strongly associated with the rate of cognitive decline in individuals with AD pathology. However, the

exact role of iron is still uncertain [4]. Additionally, an updated theorem on the role of iron by Ayton et al. [5] proposes that iron plays a bigger and more active role in neurodegeneration.

1.2 Problem

Although the role of iron in the context of Alzheimer's disease (AD) is becoming clearer, several questions remain unanswered. The role of iron in different brain regions remains unclear. The reason that we are interested in the role of iron in the *different* brain regions is that the canonical proteins amyloid and tau accumulate in a very specific temporal and spatial pattern (the Braak and Thal stages). We want to know how iron fits into this time course. Ayton et al. [4] showed that iron levels were elevated in the inferior temporal cortex in patients who were diagnosed with clinical AD. However, that is just one part of the brain.

To get a better understanding of the role of iron in the different brain regions, the behaviour of iron in these regions is being researched. From patients who are deceased with a diagnosis of Alzheimer's disease, a part of the brain is cut out, stained and scanned. Then, based on a visual analysis by experts, a score is assigned based on the amount of iron accumulation present. Comparing these scores with those from healthy controls can provide insight into the spread pattern of iron in patients with Alzheimer's Disease.

An example of such a scan can be found 1.1. In the image, we can see that the tissue has a range of brown colours. The tissue has been treated with a special chemical in a process called staining to enhance the visibility of the iron present in the tissue. The LUMC has developed a staining protocol to highlight iron [6].

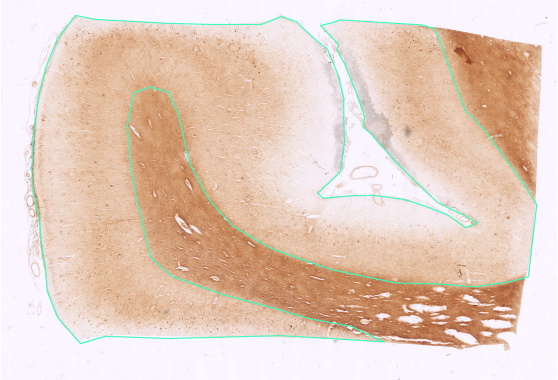

Figure 1.1: Example of brain region with iron staining and manual segmentation. The cyan border shows the manual segmentation around the grey matter

In this image, we can also see a cyan-coloured border around part of the tissue. This border encircles the grey matter. The other part of the tissue that is outside the border is the white matter. Aside from the tissue, there is also background. Both of these types of tissue are part of the brain. In this research, we focus only on the grey matter.

The scoring system for these images ranges from zero to three. An overview of example images for the four different scores can be found in Figure 1.2. In image A, we can barely see any colouring from the staining. This indicates that there is little iron accumulation. As the labels progress, more and more colouring from the staining can be seen in the grey matter. Not only is there more colouring, but what matters as well is the pattern of the iron. It starts as a band in the middle cortical layers, which can be seen quite a bit at the purple arrow in image B. As it progresses to image C, the iron accumulation spreads to the deeper cortical layers, so towards the white matter. This can be seen at the purple arrow in image C. Finally, in image D, we can see that the iron has spread to all cortical layers. It is important to note that the heads of the cortical tissue, or the head of the gyrus,

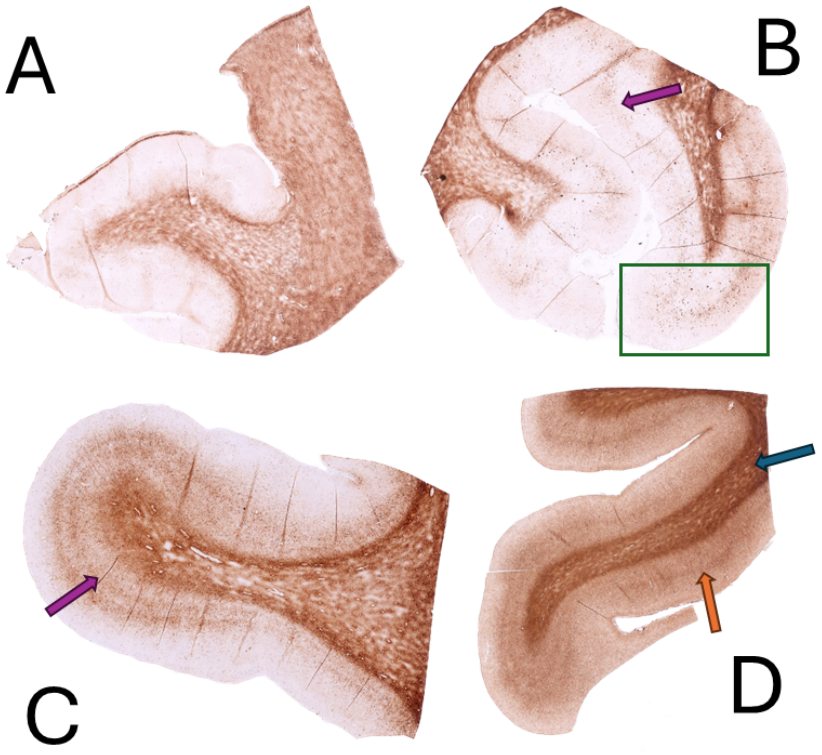

Figure 1.2: An overview of the four different scores. Labels 0, 1, 2 & 3 belong to images A, B, C & D, respectively. The two purple arrows point to iron accumulation in de cortical layers. The orange arrow points to grey matter, and the blue arrow points to white matter. The green square highlights the head of the gyrus.

where the cortex folds back into itself, are not relevant to the classification of the iron accumulation due to the complicated anatomy of the region. The head of the gyrus is highlighted with the green box in image B.

The overall staining of the tissue and the actual pattern of the iron are the two aspects that experts primarily evaluate when assigning a score to these images. However, experts may disagree about the label for the image. This problem is referred to as inter- and intra-rater consistency. These are two distinct metrics to consider. Inter-rater consistency refers to the consistency between different raters. Intra-rater consistency refers to the same rater rating the same item multiple times.

Both of these metrics are important. In an ideal world, every rater would give the same label, and the same rater would re-rate every item the same way [7].

This human inconsistency is the primary issue in labelling images. Different people may label the same image differently, and even the same person might make varying decisions at different times. One goal in applying machine learning is to reduce this variability by using models that could produce more consistent outputs under the same conditions. While machine learning models are not perfect and can have their sources of error, they typically generate the exact predictions when presented with the same data repeatedly. This consistency can help improve the reliability and reproducibility of labelling processes.

However, there are still plenty of hurdles to overcome when a machine needs to classify these images. For example, for this research, we only consider the grey matter, but the image contains more than that. So first of all, the grey matter needs to be segmented out of the scan. Additionally, the colour intensities of the images are modified to make them more visible to humans, allowing for better evaluation of the borders and colours. Maybe this is also necessary for a machine. Lastly, there are some artefacts in the images, such as tears or poor staining. Humans can ignore those elements, but it is very hard to explain to a machine how to ignore them. In conclusion, there are numerous challenges to overcome before a machine can label images as effectively as a human can.

1.3 Research Questions

This brings us to the research questions. The first research question relates to the initial step we need to take for classification, namely, extracting only the grey matter from the image, as this is the only relevant part of the tissue. As a sub-question to this, we need to ask ourselves what the necessary pre-processing steps are before proceeding to segmentation. For example, do we need the colour correction or not?

Research Question 1 *Given the iron staining of cortical brain tissue, what machine learning architecture is effective at segmenting the grey matter?*

Sub-Research Question 1.1 *Which preprocessing steps are necessary for an effective segmentation?*

After the grey matter has been segmented from the image, we can proceed to the classification. Once again, we wonder what the most effective architecture is. Additionally, we aim to determine the optimal ratio between the patch size and the downsampling rate.

Research Question 2 *Given the segmentation of the grey matter and the iron staining of cortical brain tissue, what machine learning architecture is effective at classifying iron severity?*

Sub-Research Question 2.1 *What is the best ratio between the patch size and downsampling rate?*

1.4 Thesis Structure

This thesis is divided into six chapters. Chapter 1 provides an overview of Alzheimer's disease, as it is crucial for understanding the problem. After that, the problem is introduced. Then, the research questions and thesis structure conclude the chapter. Chapter 2 gives a brief overview of the related works for the segmentation. Then, in Chapter 3, an overview is provided of the materials used in this thesis. The chapter also explains the pre-processing steps undertaken to prepare the dataset for training. In Chapter 4, the training itself is explained and all the different parameters that were used. Chapter 5 will document and analyse the results achieved for the segmentation and classification model. Finally, Chapter 6 will conclude this thesis with a brief overview, a conclusion and some thoughts on the limitations and future work.

2 Related Works

2.1 Segmentation

Segmentation is a process that aims to separate, or segment, a specific form or object from an image. In some cases, this can mean removing the background of a portrait picture of someone or in this case, segmenting tissue from histological images. In the past, segmentation was performed using edge detection methods, where individual pixels were compared with one another. In 1988, Yanowitz and Bruckstein [8] gave an interesting example of a new method for segmentation via adaptive thresholding. They considered the gradiënt of the pixel values going from one pixel to another. When the gradient exceeded a certain adaptive threshold, it was regarded as part of the segmentation. Although this works nicely for grey-scale images, things have become more complicated. Back then, pure mathematics and logic were the standard, whereas now deep learning dominates the field.

In 2015, Ronneberger, Fischer & Brox published their paper called "U-Net: Convolutional Networks for Biomedical Image Segmentation" [9]. They used a custom Convolutional Neural Network (CNN) architecture to segment images in the biomedical field. This revolutionised the field of segmentation, not only in the biomedical field but in segmentation in general. Meanwhile, the paper has gathered

more than 100.000 citations and has its own Wikipedia page. In their work, Reza et al. [10] reviewed the success of the U-Net architecture family. They evaluate 50 different architectures that all originated from the U-Net architecture. Next to this family of U-Net architectures is a self-adapting framework called nnU-Net. This is a framework created by Isensee et al. [11] that, based on your data, finds the best combination of various U-Net architectures like 2D, 3D, etc. It will configure parameters such as learning rate, batch size, and folds based on the supplied data. It has outperformed custom-made architectures based on the idea that it is better to fine-tune the training parameters rather than define new architectures. Because of this, it can be a good benchmark method for custom-made architectures.

The U-Net is a good example of a CNN. However, other architectures are also popular in the field of segmentation. Originally from the field of natural language processing, vision transformers have made their way into our fields of machine learning. Introduced in 2020 by the Google research team [12], Vision Transformers (ViT) models gained popularity initially in the classification field. But the transformer concept was soon applied to many machine learning tasks. From the U-Net architecture, there is the UNET Transformers (UNETR) that combines a U-Net with transformer blocks [13]. Another example of a U-Net architecture that was combined with transformers is the Swin U-Net [14]. The author of the UNETR has also combined these two architectures to produce the Swin UNETR [15].

Google is not the only big tech company to release segmentation models. In 2023, the Meta AI research team published a paper called 'Segment Anything' [16]. In their paper, they tackled three different tasks. The goal was to create an interactive segmentation experience. Therefore, their first task was to develop an

LLM that returns a valid segmentation for every prompt. The second task was to create the model. It begins with an image encoder based on the ViT, released by Google. Ultimately, it also includes a transformation to decode the mask. For the final task, they set out to build a data engine to handle all the images they have collected. Their state-of-the-art performance shows the importance of Vision Transformers.

2.2 Classification in histopathological images

The application of deep learning to histopathological image analysis represents a natural convergence of two key factors: the ability of deep neural networks to extract meaningful patterns from large datasets, and the abundance of high-resolution microscopy data already available in pathology laboratories. This combination has driven rapid adoption of deep learning techniques in the field, with the number of publications increasing eightfold from 300 in 2018 to 2500 in 2024 [17]. The field's foundation was established with traditional convolutional neural network (CNN) architectures. One of the pioneering works was that of Ciresan et al. [18], who applied CNNs to the detection of mitosis in breast tissue. This approach of using CNNs for patch-level classification quickly gained traction, as demonstrated in the 2017 challenge for detecting lymph node metastases in breast cancer, where all winning submissions employed CNN-based solutions [19]. Building on this success, researchers began adopting more sophisticated CNN architectures. Coudray et al. [20] successfully applied Google's InceptionV3 model to lung cancer detection, while Fu et al. [21] later used the improved InceptionV4 architecture. These works demonstrated that established computer vision

architectures could be effectively transferred to histopathological analysis. However, histopathological image analysis presents unique challenges that traditional CNNs struggle to address. Whole slide images (WSIs) are typically gigapixel-sized, making direct classification computationally intractable. Moreover, diagnostic decisions often require integrating information across multiple tissue regions rather than focusing on individual patches. These limitations led to the development of Multi-Instance Learning (MIL) approaches, which treat each slide as a collection of patches (instances) and learn to make slide-level predictions from patch-level features. Campanella et al. [22] demonstrated this concept by using CNNs to extract features from individual patches, then employing a recurrent neural network (RNN) to aggregate these features for final slide-level classification. Recent advances have further refined MIL approaches by incorporating attention mechanisms, which enable models to automatically focus on the most diagnostically relevant tissue regions. Ilse et al. [23] pioneered attention-based MIL by designing an architecture that learns to weight patch contributions based on their relevance to the final diagnosis. This approach was further developed by Lu et al. [24] in their CLAM (Clustering-constrained Attention Multiple Instance Learning) framework, which combines attention mechanisms with clustering to enhance both performance and interpretability.

Building upon attention-based MIL frameworks, CLAM's multi-branching version creates dedicated branches for each class in the classification problem, allowing the model to develop class-specific attention patterns that identify distinct morphological features relevant to different diagnostic categories.

CLAM employs an attention backbone that calculates attention scores for each patch. Rather than producing a single set of weights, it generates class-specific

attention distributions through its branching structure. Each branch functions as a specialised attention module dedicated to a particular class. For instance, in distinguishing between normal tissue, benign lesions, and malignant tumours, the model creates three distinct branches.

The attention backbone processes the same set of patch features through separate attention pathways. This parallel processing allows each branch to focus on different aspects of tissue morphology: one branch might prioritise cellular density patterns indicative of malignancy. In contrast, another focuses on architectural features characteristic of benign conditions. Each branch learns to weight patches based on their relevance to its assigned class, developing unique attention patterns that highlight class-specific morphological characteristics.

The multi-branching approach offers key advantages over single-branch attention mechanisms. It allows for more precise localisation of class-specific features, as each branch specialises in detecting particular morphological patterns without interference from features relevant to other classes. It also provides enhanced interpretability by generating class-specific attention maps that clearly show which tissue regions contributed to each diagnostic consideration.

This architecture mirrors the cognitive process of pathologists, who examine slides while considering multiple differential diagnoses simultaneously, weighing evidence for different possibilities based on distinct morphological criteria. The multi-branching CLAM framework thus provides a more sophisticated approach to automated histopathological analysis that better captures the complexity of diagnostic decision-making.

The evolution from traditional CNNs to attention-based MIL represents a fundamental shift in how deep learning approaches histopathological image analysis,

moving from patch-level pattern recognition to slide-level reasoning that more closely mimics pathologist decision-making processes.

3 Materials

3.1 Dataset

3.1.1 Overview

The dataset in this thesis consists of two cohorts. The first cohort is from the Memory and Aging Project (MAP). This is a longitudinal, epidemiologic clinical-pathologic study of dementia and other chronic diseases of ageing [25]. The second cohort is from the Netherlands Brain Bank. This is a non-profit organisation in the Netherlands that collects brain tissue from patients with various neurological and psychiatric disorders. A part of their dataset is made available for research at the LUMC [26].

In the MAP cohort, 40 cases are included. Each of these cases has a duplo scan made for five brain regions. These regions are the occipital, inferior temporal, mid-frontal, parietal and cingulate cortex. An overview of the brain regions can be found in Figure 3.1. For the NBB cohort, 34 cases are included. Most cases also have a duplo scan for four brain regions. These are the same regions as for the MAP cohort, but without the cingulate. However, not every case has a scan for each brain region, and sometimes there are more than two scans per brain region.

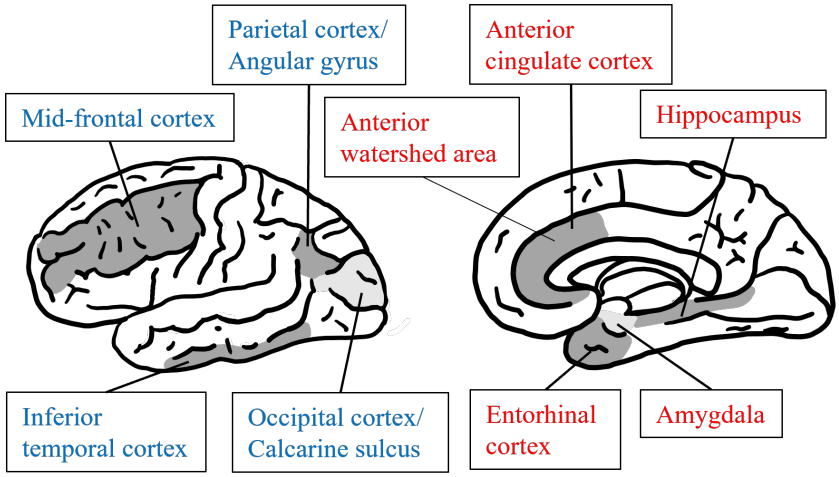

Figure 3.1: Overview of the relevant brain regions

	Iron Scored		Segmented	
Brain regions	MAP	NBB	MAP	NBB
Occipital	72	41	8	23
Inferior Temporal	78	69	7	56
Mid-frontal	77	37	11	25
Parietal	78	66	8	34
Cingulate	77	-	6	-
Total	382	223	40	138

Table 3.1: Amount of scans (Iron scored and Segmented) per cohort per brain region

Some cases have been excluded from both cohorts.

Each scan that was not excluded has received an iron score. However, not every scan has been segmented due to the time intensity of the work. A complete overview of the amounts per cohort per region is provided in Table 3.1.

3.1.2 Technical Details

These scans were performed using the Ultrafast Philips slide scanner at a resolution of 0.25 μm. The original size of the tissue is approximately 2 cm by 2 cm. This means that the scans at full resolution have 80.000 by 80.000 pixels. When the files are uncompressed, they can be as large as 13 GB. The scans are stored as pyramidal TIFF files to accommodate their large size. These pyramidal TIFF files contain the scan at increasing downsample sizes, ranging from the original size of 1x to 256x downsampled. This makes it easy to retrieve a downsampled version of the original scan quickly.

3.2 Pre-processing

This section explains the various pre-processing steps undertaken to prepare the data for use. There are three distinct sections: general pre-processing, pre-processing for segmentation, and pre-processing for classification.

3.2.1 General

Cropping

The scanner automatically determines the region of interest. However, this process does not always correctly determine that region. This means that sometimes the scanner includes more area around the actual tissue than is desired. An example of such an inclusion of extra area around the tissue can be seen in Figure 3.2. Here, we can see the tissue and the region of interest on the right. Next to that, there are two smaller squares that don't contain any tissue. These are mistakenly

Figure 3.2: The original image of a scan that includes the region of interest, two smaller regions not of interest and a large red part that was used as filler by the scanner

included in the image. To fill in the image, the scanner includes all this red area, making it one image instead of three separate ones.

To combat this problem, a quick algorithm was designed to crop these images. There are five steps in this process. First, the image is converted to a greyscale image. Second, the image pixels are adjusted with an OTSU threshold. This function in OpenCV automatically determines the correct threshold values to set the pixels to either 0 or 255. Third, this image is inverted because the fourth step requires this transformation. Fourth, a connected components analysis is done. Once again, this is a function provided by OpenCV. This function returns the leftmost pixel, the topmost pixel, the area, the width and the height. The area is used to determine the largest area. However, it is essential that this area does not have the width and height of the original image. As shown in Figure 3.2, if this constraint is not implemented, the red area would be the largest, and the image would not be cropped. For the fifth step, the image is cropped. A visual representation of each step can be found in Figure 3.3.

Colour Correction

As briefly mentioned in the Problem Statement 1.2 and Research Questions 1.3, colour correction is a pre-processing step that is necessary for humans and may also be required for a machine. The original colour correction to analyse the images

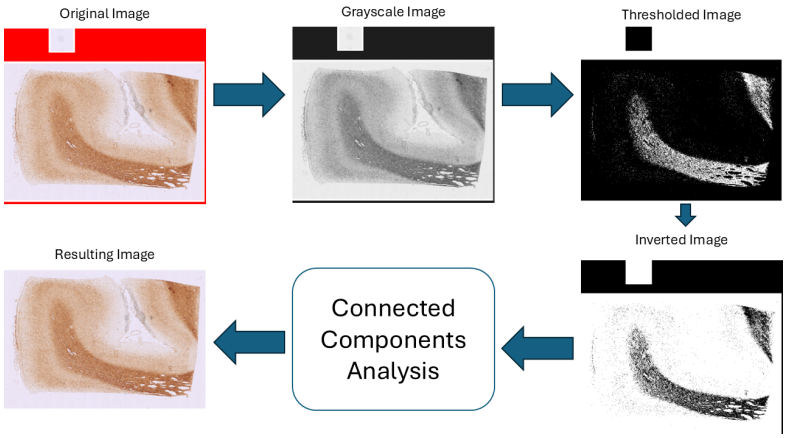

Figure 3.3: The five steps of the cropping process

was done in QuPath, but this was not saved. Therefore, this had to be applied to the images before training as well.

For each image, an expert has determined the minimum and maximum channel values, ensuring that contrast is improved while the image remains readable. In Figure 3.5, we can see how these values were determined. The pixel values in the green square became the new maximum channel values, and the pixels in the blue square became the new minimum channel values. For some images, this also meant that the contrast did not need to be adjusted at all, as it was already sufficient. These channel values refer to the RGB channels. Rather than defining new values for each channel, the image is first converted from RGB into a greyscale image, and then the new channel value is determined.

To modify the channel values, first, the minimum and maximum channel values are taken as input. Secondly, every pixel that is above the maximum channel value is changed to the maximum value. Every pixel below the minimum value is set to 0. Third, the minimum channel value is subtracted from all pixel values, such that the range the pixels occupy is shifted down to 0. Lastly, this range is then scaled

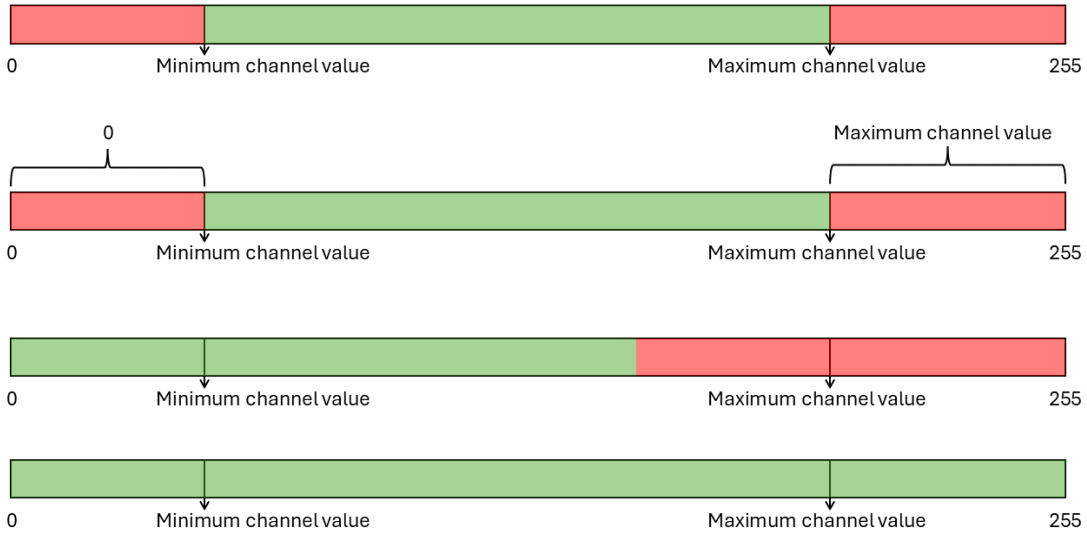

Figure 3.4: A visual explainer of the colour correction

to occupy the full 0 to 255 range. This way, the new minimum value represents 0, and the maximum value represents 255. A visual explainer of this process can be found in Figure 3.4. In Figure 3.5, the effect of the colour correction can be seen.

3.2.2 Segmentation

In the general section, the pre-processing steps that were necessary for both classification and segmentation are described. This section explains the pre-processing step required for segmentation, specifically downscaling.

The manual segmentations were made on a 20x downscaled image. The problem, however, was that these images were saved as JPEG files, and the framework used for segmentation did not accept JPEG images. We felt that it was not sufficient to convert these JPEGs to PNG images because the images might lose quality

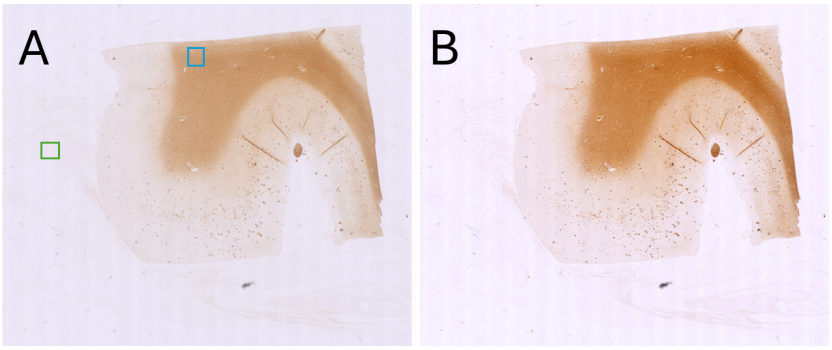

Figure 3.5: An example of the colour correction. Image A is before the colour correction and Image B is after the colour correction. The green square shows the area that determined the maximum channel value. The blue square shows the area that determined the minimum channel value.

due to JPEG being a lossy compression algorithm. Therefore, the images had to be recreated and saved as PNGs. The pyramidal TIFF formats allowed for easy downscaling, as a 16x downscale was readily available. Then the new size from the 16x downscale had to be calculated, and the image needed to be rescaled to the new size.

The segmentation masks did not fit immediately on these downscaled images because they were created from images that had not been cropped. This means that the segmentation masks needed to be adjusted to the new size of the images. This was achieved by shifting the segmentation masks to match the extent of the original images' cropping.

3.2.3 Classification

This section explains the four pre-processing steps required for classification. Two of the four pre-processing steps were required by the framework used for the classification. The other two steps are a result of this setup.

As mentioned earlier in Research Question 1.3, the segmentation model's task was to segment the grey matter from the entire image, allowing it to be used for classification. However, the segmentation model only draws the border on the grey matter but does not extract it. Therefore, the first two pre-processing steps will extract the segmentation from the original images.

The first step in this process is to upscale the mask that the segmentation model produced. The segmentation model creates segmentation masks for the 20x downscaled images. However, that is not the resolution we want to use for the classification.

Expert rating of iron accumulation is based on the pattern of iron distribution over cortical layers, as well as punctate iron accumulation in amyloid plaques and activated glial cells. Given the size of these features, a combination of 4x downsampling and 256x256 patch size was visually selected as the most optimal to contain these typical patterns within a single patch. In Figure 3.6, we can see two patches selected from an image where these two features, namely the plaques and glial cells, fit in the patch size. The black dots with small spikes in the purple square represent the glial cells. They are smaller than the plaques, but sometimes they occur next to each other, like in this patch. In the blue square, we can see an amyloid plaque. Here, the iron in the amyloid plaques that combine is represented by this large coloured mass.

Therefore, the mask generated by the segmentation model needs to be upscaled to fit that resolution. This is achieved by calculating the new image size that the masks should fit and resizing them accordingly. The interpolation method used in this case involves examining the nearest pixels. The reason behind this interpolation method is that all other interpolation methods offered by the OpenCV resize function are made for pixel gradients. For example, linear interpolation interpolates the pixel value between the pixel values on the left and the right. This does not work for a binary mask since the values can only be 0 or 1.

The second step in this process is to extract the grey matter from the image using the upscaled mask. Since the segmentation mask is binary, we can use a bitwise AND function to extract only the pixels from the image where the segmentation mask has a 1.

Now that a segmentation mask is available, two additional pre-processing steps are required by the framework. The third step is creating patches. The critical part of this step is the patch size. This is one of the major hyperparameters that we can adjust in this setup to enhance the model's performance. When it was determined that 4x downsampled was the best option for the classification, this was determined using the patch size of 256 x 256 pixels. This is also the standard for the framework when operating on scans, such as the ones used here. The way this patch size was determined was by looking at the size of the region of interest that the model should recognise, as determined by the experts. However, to ensure that the model captures bigger amyloid plaques, which sometimes occur, we also decided to use a patch size of 512×512 pixels. The framework handles the actual generation of the patches.

Figure 3.6 gives a visual representation of how these patches are extracted. This

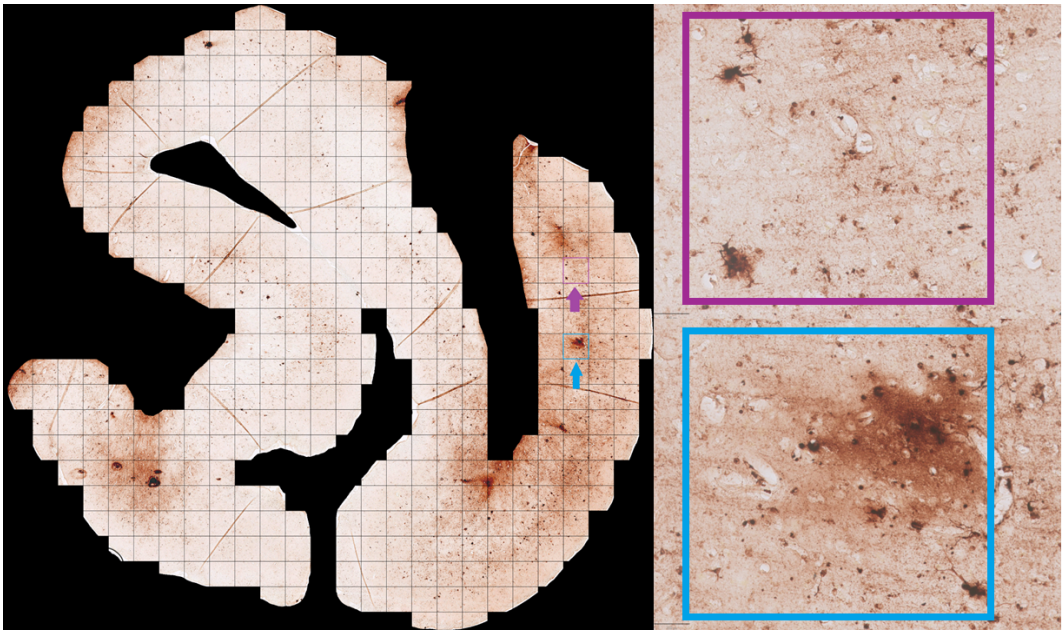

Figure 3.6: An image that is stitched together that shows the patches. This image was downscaled by a factor of 8 with a patch size of 256. The black dots in the purple square are the glial cells, and the darker mass in the blue square is an amyloid plaque.

image was downscaled by a factor of 8 with a patch size of 256. A grid is drawn over the image to show the origin of the patches. The framework produces these images for every image that was patched. This means that we can inspect every image to verify that all previous steps were completed successfully, and potentially remove or redo some images that were not processed correctly. Additionally, we can see how the framework handles patches that are on the edges. The framework excludes patches whose centre does not lie within the borders of the segmentation.

In their research, Peng et al [27] also classify features on WSIs. They determined that, given a patch size of 1024 x 1024, the region of interest was completely captured at a 4x downsampling rate. However, they achieved the best results at

8x downsample. Therefore, we also decided to include 8x downsampled data in the experiments.

The fourth step in the process is the feature extraction. The framework utilises a pre-trained ResNet-50 encoder to encode the patches into a 1024-dimensional feature vector. The ResNet-50 encoder is the standard encoder when using the framework. The resulting feature vector will be the input for the classification model. The goal of this step is to reduce dimensionality, capture useful representations, and prepare for downstream tasks. Reducing the dimensionality of the patches to the features has two results. It saves disk space and reduces the necessary network architecture to process feature vectors instead of patches.

4 Experimental Setup

4.1 Segmentation

4.1.1 Data Partitions

A total of 167 cases were manually segmented. From these 167 cases, five have been removed due to poor quality. Therefore, a total of 162 manually segmented cases were identified. To adhere to a commonly used split of 70% training cases, 10% validation cases and 20% testing cases, 35 cases have been split off for testing purposes. With the remaining 127 cases, five hand-made splits have been made for cross-validation. They have been made manually for two reasons. The first reason is to maintain even splits across all training runs. Now, only the model hyperparameters are changing. This way, we can adequately evaluate changes to the model, rather than relying on a random split that just happens to be a good one. More importantly, the second reason they were made by hand is to prevent data leakage. Every case has a duplo of scans of the same region. The duplo's are not the same, but similar enough for the model to consider them equal. This would give unfair and unrealistic results. Thus, all images belonging to a single case appear in either the training, validation, or testing set during a single training

Fold	Training	Validation
0	101	26
1	101	26
2	101	26
3	101	26
4	99	28

Table 4.1: The number of images per fold

Label	Count	Area
0	19	3.32e+06
1	38	3.75e+06
2	56	4.14e+06
3	49	3.71e+06

Table 4.2: The number and area of images per label

run. An overview of the total number of images per fold is provided in Table 4.1. The number of images in the folds is not the same because there is an imbalance in the number of scans per case, and it is beneficial to keep the regions distributed evenly.

Next to that, remark Table 4.2. This table shows an overview of the number of images per label and the average area per label. Not every label is represented evenly due to the natural occurrence of these labels. Some labels are more frequent than others. It also shows the average area per label in pixels. There is not a vast difference in area, but it may be significant enough to impact the model's performance. The model should generalise as much as possible, and that also means that the performance should be independent of the area.

Lastly, it is essential to note that this setup has been run twice, as described in Section 3.2.1, once with colour correction and once without colour correction. This way we can analyse whether the colour correction affects the performance of the segmentation model.

4.1.2 Model

The exact version of U-Net used in this thesis is called a Residual U-Net [28]. The reason for this exact version is that the residual version outperformed the original architecture [29]. The residual part refers to a modification to the original U-Net architecture. In the encoder blocks, it uses the residual network blocks rather than the original convolution blocks. This strategy tries to incorporate the strengths of the residual blocks in the U-Net architecture. An overview of the two differences can be seen in Figure 4.1. Support for Residual U-Net was added to nnU-Net in 2024 [29].

This model uses a Nesterov version of the stochastic gradient descent optimiser. For the learning rate, it uses a polynomial learning rate scheduler with the following formula:

$$\text{new\_lr} = \text{self.initial\_lr} * (1 - \text{current\_step} / \text{self.max\_steps}) ** \text{self.exponent}$$

Here, the max_steps is the number of epochs it plans to run, and the exponent is set to 0.9.

In Table 4.3, an overview can be found of the relevant packages and their versions used for the training of this model. Training was performed on an NVIDIA L40 GPU.

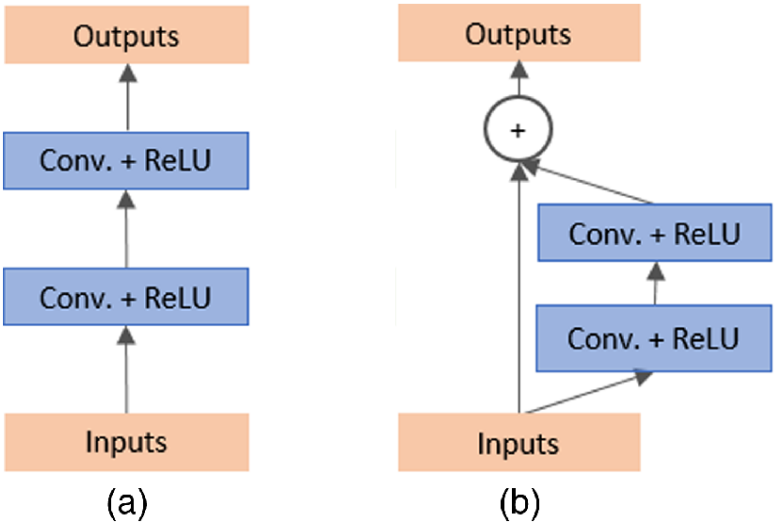

Figure 4.1: Different variants of convolutional and recurrent convolutional units (a) Forward convolutional units, (b) Residual convolutional units [28] (Conv is a convolutional layer and ReLU is the Rectified Linear Unit activation function)

Software Package	Version
python	3.10.12
cuda	11.8.0
nnUNetv2	2.6.0
torch	2.6.0+cu118
dynamic-network-architectures	0.3.1
batchgenerators	0.25.1
pillow	11.1.0

Table 4.3: The relevant software packages used and their versions

Parameter	Value
$initial\_lr$	$1e-2$
$weight\_decay$	$3e-5$
$num\_iterations\_per\_epoch$	250
$num\_val\_iterations\_per\_epoch$	50
num_epochs	200

Table 4.4: Segmentation Model Hyperparameters

4.1.3 Training

Most of the training hyperparameters are determined automatically by nnU-Net. It could be worth fine-tuning some of the parameters manually when a hyperparameter does not align with our prior knowledge. However, this might not be time well spent as nnU-Net most likely decides the best hyperparameters. A Z-score normalisation is used for normalising the images. NnU-Net is not designed to automatically determine every hyperparameter based on the provided dataset, because they are mostly independent of the dataset. Some are manually set to the number that is deemed optimal in most situations. These can be found in 4.4. The only hyperparameter that has been tweaked in this thesis is the number of epochs. The code has been modified to store checkpoints at 50-epoch intervals, allowing for evaluation of whether the model is overfitting.

4.1.4 Metrics

For evaluating the segmentation model, two metrics have been chosen to assess its performance. The first metric is the Dice score (also known as Sørensen-Dice)[30][31]. This is an overlap metric, which means it examines how well the prediction aligns with the mask. Dice first wrote it down as:

$$\frac{hn}{ab} \tag{4.1}$$

Where a is the number of elements only in class A or also in B, b is the number of elements only in class B or also in class A, h is the number of elements in both class A and B, and n is the total number of elements[31]. This is analogous to the more common representation:

$$\frac{2|A \cap B|}{|A| + |B|} \tag{4.2}$$

where $|A|$ refers to the cardinality, or the length, of the set of elements in A, similarly for B. $|A \cap B|$ denotes the intersection of elements in both A and B.

The dice score is widely used for segmentation. It is easily interpretable because it is a score between 0 and 1, where 0 represents the worst and 1 represents the best, and it is straightforward to compute. The reason it is used in this case is that this problem is a straightforward semantic segmentation. There are no small objects to detect, and most of the masks have a similar size. There is, however, one downside to this version of the Dice scores. It has a positive bias towards the size of the ground truth mask, i.e., a higher number of pixels in the mask will lead to a higher Dice score [32]. Raina et al [32] propose a scaling factor at the recall rate to reduce the bias towards the fraction of positive pixels in the mask.

Consider Equation 4.2 in this binary segmentation problem. We can rewrite this equation in terms of True Positives (TP), False Positives (FP) and True Negatives (TN) to

$$\frac{2TP}{2TP + FP + FN} \tag{4.3}$$

since $|A \cap B|$ accounts for the pixels from the prediction B that belong to the ground truth mask A, or the TP pixels. $|A|$ accounts for the TP pixels but also the FP pixels, and $|B|$ accounts for the TP pixels and the FN pixels. Combined, the resulting equation is 4.3. Finally, this can be rewritten with some terms substituted to

$$\frac{2}{2 + p + n} \tag{4.4}$$

where $p = \frac{FP}{TP}$ and $n = \frac{FN}{TP}$. The scaling factor, as mentioned above, is calculated by

$$h(r^{-1} - 1) \tag{4.5}$$

where h is the ratio between the number of positive pixels and the number of negative pixels. The r is the average lesion load for all the images. Lesion load, in this sense, refers to the average number of positive pixels. Then, this scaling factor is applied to the p to yield the nDSC.

The nDSC will be the reported metric for segmentation because it is less biased towards the size of positive labels. The pseudo dice that nnU-Net reports in the graphs they produce of the training is still the normal Dice score. However, it is called pseudo because it is the Dice score of the validation set. The idea is that this approximates the Dice score for the eventual test set, hence the term 'pseudo

Dice'.

$$nDSC = 2(2 + kp + n)^{-1}, k = h(r^{-1} - 1) \tag{4.6}$$

Next to an overlap metric, Maier et al. [33] recommend including a boundary-based metric, such as Hausdorff distance or Normalised Surface Distance (NSD), for semantic segmentation problems. The human-annotated masks for the segmentation are not entirely perfect. It does not make sense to strictly adhere to that segmentation for evaluating the model. In such cases, an NSD metric can be included to correct for imperfections in the masks [33].

The NSD is a metric that compares how well the boundaries overlap between the prediction and the mask. Around the border of both masks, there is a boundary region. The amount of border from the mask that falls within the boundary of the prediction and vice versa is added and then divided by the total border area. See Figure 4.2 for a schematic overview of this metric.

A tolerance parameter was chosen with consultation of the experts who label the images. It has been determined that a tolerance of 25 pixels around the border is a suitable option, as the imperfections in the manual segmentation fall within this 25-pixel range. The tolerance cannot be too high, as this would result in the entire segmentation being in the overlap, and the value of the NSD score would be lost. A visual of what this boundary looks like can be seen in Figure 4.3.

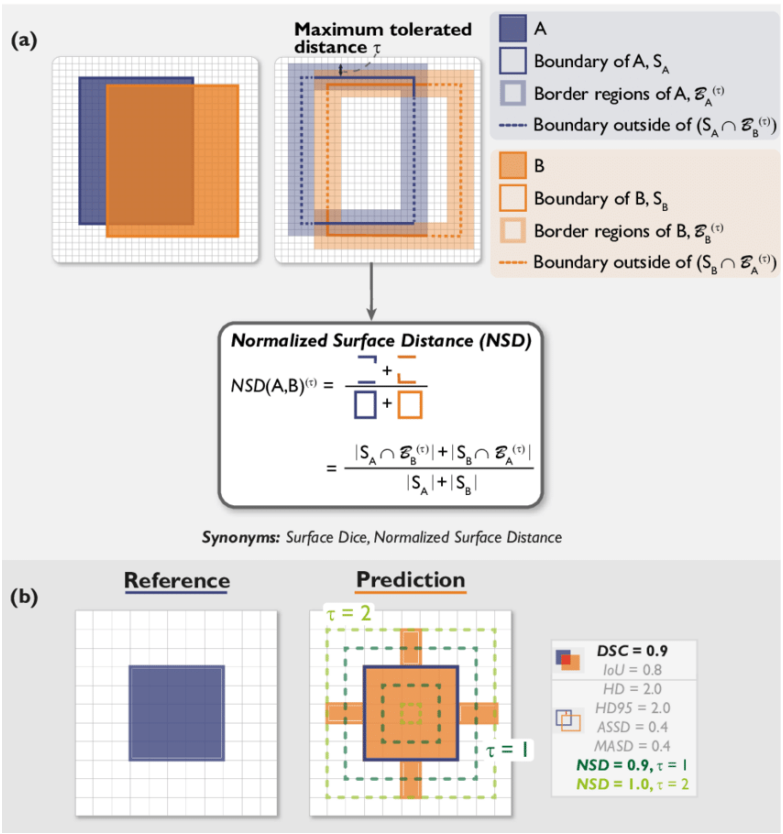

Figure 4.2: Explanation of the NSD metric [34]

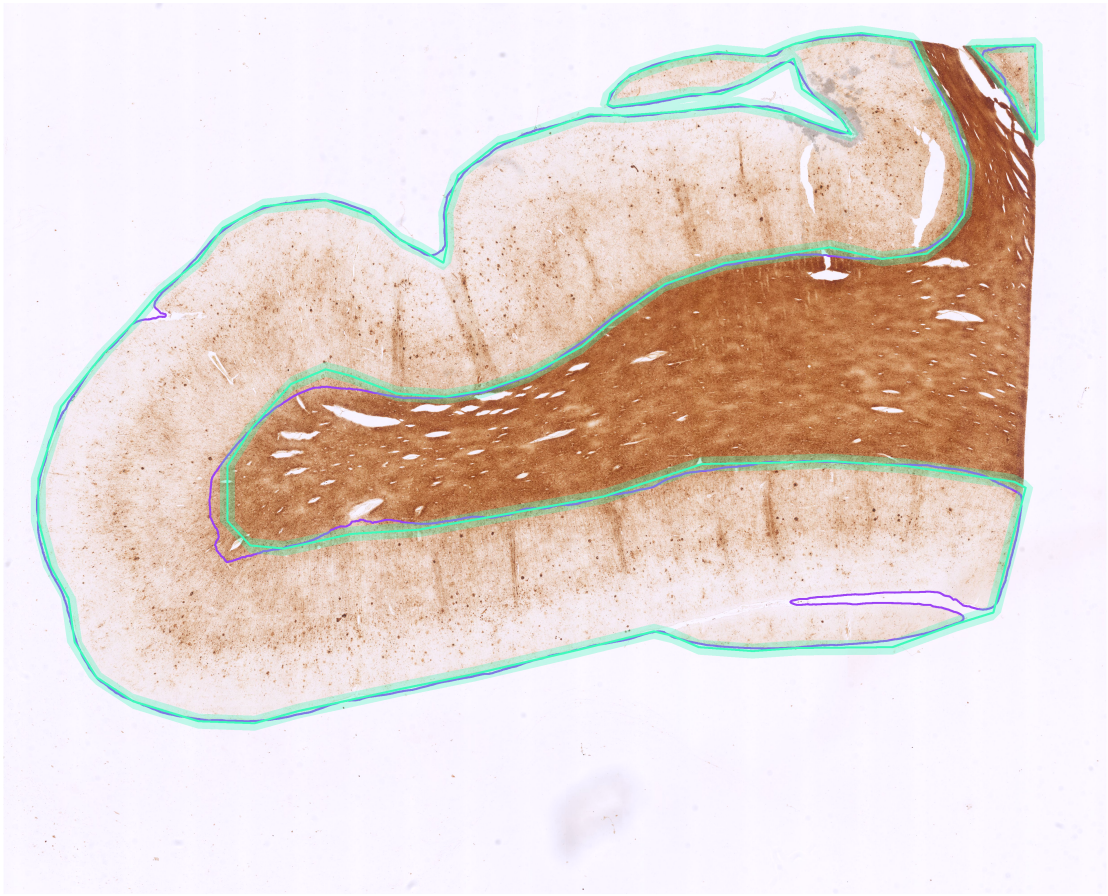

Figure 4.3: An image with the 25 pixel tolerance boundary around the gold standard. The cyan colour shows the gold standard and the purple shows the prediction from the model.

Level	Label 0	Label 1	Label 2	Label 3	Total
Patient	9	16	27	32	64
Image	14	21	40	52	127

Table 4.5: Amount of images and patients per label for the Inferior Temporal brain region. The individual patient counts included per label do not add up to the total number of patients because some patients appear in multiple labels.

4.2 Classification

4.2.1 Data Partitions

Due to the size of the files at their original size and the size of the pre-processed files, we chose to verify the architecture and setup using data from the Inferior Temporal brain region only. The total number of images that were used was 127. Recall Table 3.1, where 147 images belonging to the Inferior Temporal region were given an iron score. In theory, all of those images were usable for the classification. However, in practice, some images could not be pre-processed. The exact cause of this technical issue remains unknown. The images that could not be pre-processed have been left out.

Similar to the segmentation in 4.1.1, the images per fold are split on the patient level. The total amount of images used per label can be found in Table 4.5.

As explained in Section 3.2.3, there are two hyperparameters when it comes to the data. There are downsampling and patch size considerations. The downsample factor will be either 4x or 8x, and the patch size will be either 256 x 256 or 512 x 512, as was determined in Section 3.2.3. There is one more hyperparameter that will be tweaked in this experiment, and that is the colour correction. This results in the following eight configurations for the classification, as shown in Table 4.6.

Experiment	Downsample	Patch Size	Colour Correction
1	4x	256x256	Yes
2	4x	256x256	No
3	4x	512x512	Yes
4	4x	512x512	No
5	8x	256x256	Yes
6	8x	256x256	No
7	8x	512x512	Yes
8	8x	512x512	No

Table 4.6: Data Configurations for the Classification Training

4.2.2 Model

The model used for the classification is the CLAM-MB model [24]. CLAM is a Clustering-constrained Attention Multiple Instance Learning method designed for classifying WSIs. More specifically, the multi-branching version is used.

4.2.3 Training

Although CLAM does not automatically determine the correct hyperparameters like nnU-Net does, numerous hyperparameters still need to be tuned. However, most of the hyperparameters have been kept at the value recommended by the authors of CLAM. An overview of the relevant hyperparameters can be found in Table 4.7. The dropout parameter is used to drop a certain number of connections after a fully connected layer, thereby avoiding overfitting. In this case, that means that 25% will be dropped after a fully connected layer. For CLAM, a hyperparameter is the *maximum* epochs. The reason for the maximum amount, rather than a definite amount, is that CLAM has a built-in early stopping mechanism. If the validation loss has not decreased in 20 epochs, the model will stop. This way, it will automatically stop overfitting.

Hyperparameters	Value
max_epochs	200
$initial\_lr$	2e-4
$weight\_decay$	1e-5
$drop\_out$	0.25

Table 4.7: Classification Model Hyperparameters

4.2.4 Metrics

The metrics for classification are accuracy and AUC. The accuracy is defined as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

However, it represents the average accuracy calculated from all the trained folds.

The AUC is the Area Under the Curve of the ROC Curve. The Receiver Operating Characteristic (ROC) curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR). The TPR is calculated by

$$TPR = \frac{TP}{TP + FN}$$

and the FPR is calculated by

$$FPR = \frac{FP}{FP + TN}$$

Plotting the TPR against the FPR shows how good the TPR can get before the FPR increases beyond a desired boundary. This desired boundary depends on the use-case. The ideal scenario would be a square graph where the TPR would be 1 when the FPR is 0. When a straight diagonal line exists in a binary classification problem, the model's performance is equivalent to guessing 50/50.

For the classification problem in this thesis, there is no binary classification problem, but a multi-class classification problem. This means that it is not possible to produce this plot directly. That is why the AUC is modified to be a one-vs-rest AUC. This means that the AUC is computed by treating one class as a positive case and the remaining classes as negative cases. Then the average is taken over the n different AUC values.

5 Results

5.1 Segmentation Results

5.1.1 Evaluation of the scores and metrics

This chapter will give an overview of the segmentation results. A significant portion of this analysis will be numerical, as it is the most tangible approach. In reality, these segmentations need to be accurate enough to be used for classification. The numbers provide a good indication of how well the segmentation has performed, but the results were also visually checked by an expert.

A comprehensive overview of the average Normalised Dice and NSD scores for the test set, including both colour and non-colour correction, is provided in Table 5.1. It can be seen that the Normalised Dice scores are all above 0.86. An overview of the spread of the scores is provided in Figure 5.1. From this figure, it can be concluded that not only is the average performance good, but the 25th and 75th quartiles are also above 0.85. There are only a few outliers below 0.80, but even these are still above 0.70. When comparing these results across the two datasets, i.e., the colour-corrected dataset and the non-colour-corrected dataset, there is little to no variation. The only differences in scores are found in the

second decimal. This indicates that the colour-correction is not relevant for the segmentation.

The pattern we see for the Normalised Dice scores is quite comparable to the pattern for the NSD scores. These scores are slightly lower than the Normalised Dice scores, ranging from 0.821 to 0.948, but most scores fall within the 0.85 range. The reason for the higher average score for label 0 is that there is only one test case, and that case just happened to have a higher-than-average score. There are also minimal differences between the two datasets. The large difference between the NSD scores and the Normalised Dice scores is due to the greater spread of the NSD scores, as shown in Figure 5.2. The NSD scores exhibit significantly more variability, with some dropping below 0.7. When comparing the highest and the lowest score for the non-colour corrected dataset in Table 5.1, the scores are still all above 0.8.

In Figure 5.3, there are two cases from the non-colour corrected dataset presented for visual inspection with their corresponding NSD scores. On the left is the worst case with a score of 0.68, and on the right is the best case with a score of 0.97. The two images have two borders around them. The cyan border is made by the human annotator. This border also has a 25-pixel error margin to indicate the border area used in calculating the NSD score. The purple border is the prediction made by the segmentation model. The left case has segmented some of the tears and folds in the tissue that the human annotator did not include. However, this does not mean that it is bad to include these pieces of tissue. This means that, although it received the lowest score due to the inclusion of tears, the rest of the segmentation is still good. In the best case, we can see that the model has predicted near-perfect borders compared to the human annotator and the tissue.

When examining both the Normalised Dice and NSD scores for both the colour-corrected and non-colour-corrected datasets, we observe a relatively narrow spread across the four labels. This is a significant result because it demonstrates that the segmentation model is agnostic to the label, even though the boundaries between images from labels 0 and 3 may differ significantly. For some images, the boundary is more visible than for others. To see the complete distribution for each of the labels for the Normalised Dice and NSD scores for the two datasets, have a look at the appendix A.1.

We also investigated whether the segmentation performance differed per brain region.. In Table 5.2, an overview can be found when the Normalised Dice and NSD scores are split by brain region. The same pattern that can be seen in Table 5.1 repeats itself here again. The Normalised Dice scores remain relatively stable, showing minimal variation across the different brain regions and consistently averaging around the same value. When examining the NSD scores, they reveal a bit more. We can see that the parietal region underperforms compared to the other areas. However, the scores of all regions stay above 0.80.

The final feature we can split the scores on is the size of the region. In Figure 5.4, a plot can be found where the area of the tissue is plotted against the Normalised Dice score. This is a check to verify that the performance of the segmentation is not correlated or somehow connected to the area of the tissue it has segmented. Ideally, it should not be connected, and as can be seen in the image, there is little to no correlation between the area and the segmentation score.

	Colour Corrected		Non Colour Corrected	
Subset	**Normalised Dice Score**	**NSD Score**	**Normalised Dice Score**	**NSD Score**
All	0.858	0.849	0.866	0.851
Label 0	0.876	0.951	0.886	0.948
Label 1	0.857	0.894	0.871	0.886
Label 2	0.867	0.837	0.869	0.835
Label 3	0.833	0.825	0.850	0.851

Table 5.1: The Normalised Dice and NSD scores for the test set per label

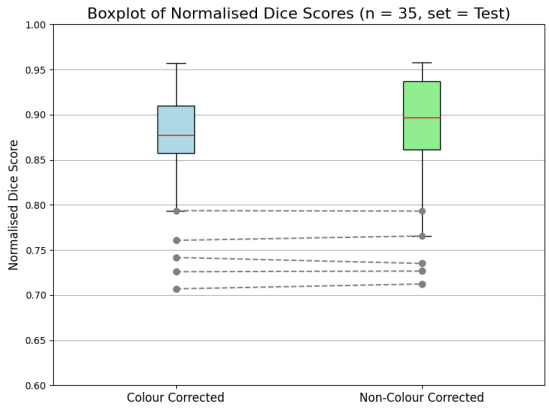

Figure 5.1: The Normalised Dice scores for the colour corrected vs the non-colour corrected set

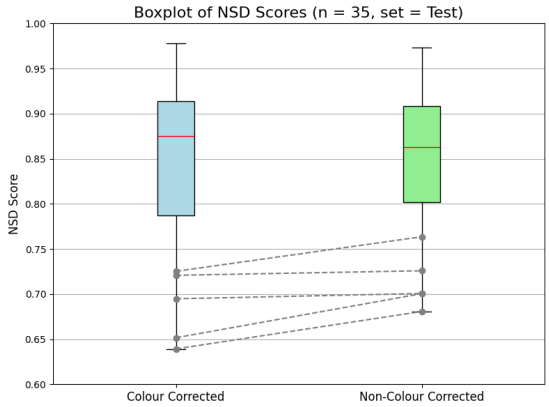

Figure 5.2: The NSD scores for the colour corrected vs the non-colour corrected set

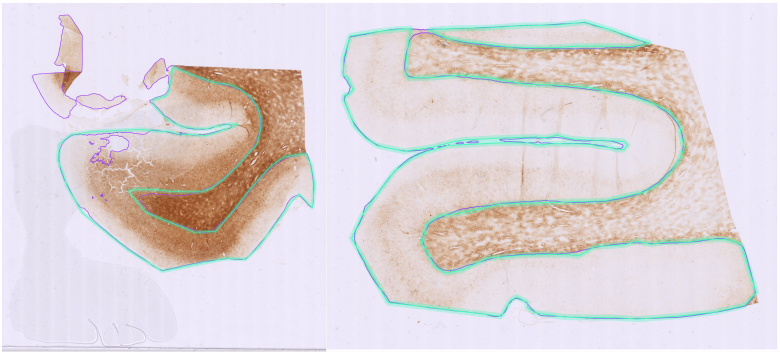

Figure 5.3: The worst (left, 0.68) and best (right, 0.97) NSD scores

	Colour Corrected		Non Colour Corrected	
Subset	**Normalised Dice Score**	**NSD Score**	**Normalised Dice Score**	**NSD Score**
Frontal	0.890	0.836	0.912	0.840
Temporal	0.837	0.873	0.840	0.850
Parietal	0.860	0.803	0.871	0.805
Occipital	0.910	0.869	0.914	0.877
Cingulate	0.834	0.861	0.835	0.895

Table 5.2: The Normalised Dice and NSD scores for the test set per region

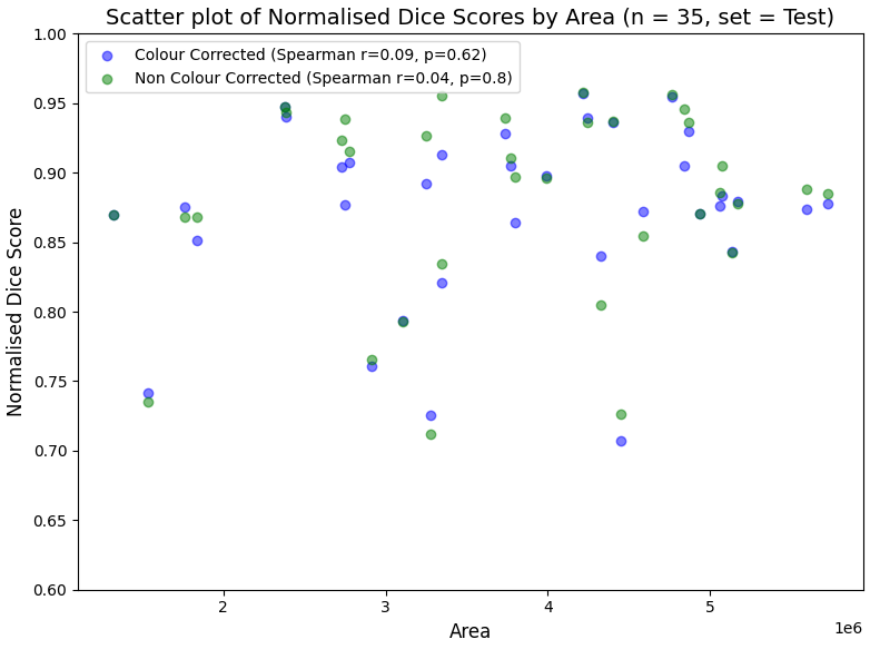

Figure 5.4: The area of the tissue plotted against the Normalised Dice score

5.1.2 Evaluation of overfitting

Overfitting is always a problem when using machine learning. The goal is to generalise the model as much as possible, such that it can also perform well on unseen cases. This section will assess how much the segmentation model may have overfitted.

During training, NnU-Net produces a graph similar to the one in Figure 5.5, where it plots the training and validation losses. This can be very insightful to evaluate overfitting. Generally speaking, when the validation loss increases but the training loss decreases, the model is likely overfitting. The validation loss starts increasing again around the 100-200 epoch mark, while the validation Normalised Dice score barely increases. This meant that 1000 epochs was too much, but it does not immediately tell how many epochs could be the optimal amount.

Although subjectively, a decision was made to train the segmentation model and save the model at intervals of 50 epochs, so at 50, 100, 150 & 200 epochs. The Normalised Dice scores for the segmentation model at 50-epoch intervals are shown in Table 5.3. The Normalised Dice scores go down by 0.02 as the epochs increase. Although they go down, which is interesting, the difference is too small to be significant. Typically, we would expect the Normalised Dice scores to grow with more training. The performance does not improve, but the model continues to learn from the cases, which could be harmful. This is harmful because the model is no longer generalising and is overfitting on the training dataset. This is not what we want.

The NSD scores increase by no more than 0.05 between the 50th and 200th epochs. This suggests that the 200-epoch variant is slightly better than the 50-

epoch variant. The Normalised Dice and NSD scores do not agree in the sense that Normalised Dice scores decrease, but the NSD scores increase as the number of epochs trained increases. Since the difference is so slight, it could also be a chance finding.

An expert review has been introduced to give more information about which epoch variant of the model performs best. Four segmentations have been made for nine randomly selected images. These four segmentations correspond to the four saves that each model has made, namely, at 50, 100, 150 and 200 epochs. Two experts have reviewed these four segmentations and have selected the best segmentation. The results can be found in Table 5.4. The results in this table show which segmentation was preferred. However, in most cases, all of the segmentations were acceptable.

Upon examining the scores, we observe a slight preference for the 50-epoch save. In some instances, the 200-epoch is the better version. The fact that the 50-epoch version is the most favourable does match the theory. If the Normalised Dice score does not improve within 150 epochs, then the model is likely overfitting and no longer generalising. If that is the case, we should stop training at 50 epochs, which agrees with the fact that the 50-epoch variant was most preferred.

In conclusion, the segmentation model performs well, achieving an average Normalised Dice score of approximately 0.85 for all tested brain regions, regardless of the staining intensity in the histology sections.

	Color Corrected		Non Color Corrected	
Epoch	**Normalised Dice Score**	**NSD Score**	**Normalised Dice Score**	**NSD Score**
200	0.858	0.849	0.866	0.851
150	0.850	0.851	0.871	0.848
100	0.857	0.846	0.866	0.842
50	0.873	0.814	0.882	0.800

Table 5.3: The Normalised Dice and NSD scores for the test set per interval

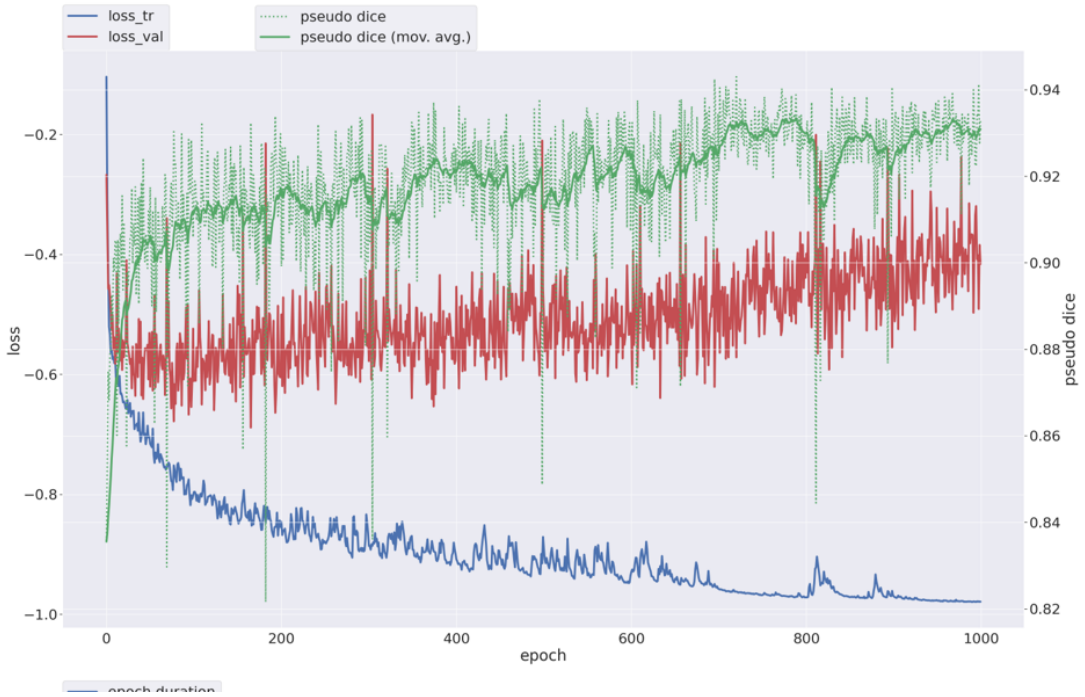

Figure 5.5: The train and validation loss for the first experiment using 1000 epochs on nnU-Net. The blue line is the training loss, the red line is the validation loss and the green line is the Normalised Dice score of the validation. It is called pseudo because it is not from the test set but from the validation set.

Slide ID	Expert 1	Expert 2
2015-035_OCC_15	150	150
74587057_1_CAL	200	200
E14-50_T_3	50	50
E14-50_F_15	100	50
ANW746_OCC_15	50	50
82478522_1_AG	50	50
51791453_1_MF	150	150
2015-009_P_17	50	-
29821047_1_CG	100	100

Table 5.4: Epochs of the best model for the segmentation chosen by the experts

5.2 Classification Results

5.2.1 Planned Experiments

This chapter will give an overview of the results for the classification model. The results from the planned experiments will be shown and discussed.

An overview of the accuracy and AUC scores for the planned experiments, as mentioned in Table 4.6, is shown in Table 5.5. The accuracy is very bad. The model is unable to predict these cases. When looking at Figure 5.6, we can see that the model does not know how to differentiate the classes from each other. The only reason the accuracy is still 0.45x is that there are many cases with label 3, and it predicted label 3 for every one of them. Although this figure only shows the confusion matrix for this experiment, they all look alike. The accuracy for each of the experiments also remains constant, except for experiment 2, where it decreases by 0.02.

Regarding the AUC, the results are also poor. In a binary classification problem, when the AUC is below 0.5, it means that the model performs worse than guessing. Although not directly applicable, it does suggest that an AUC of around

Experiment	Downsample	Patch Size	Colour Correction	**Accuracy**	**AUC**
1	4x	256x256	Yes	0.450	0.433
2	4x	256x256	No	0.431	0.445
3	4x	512x512	Yes	0.451	0.430
4	4x	512x512	No	0.451	0.423
5	8x	256x256	Yes	0.451	0.511
6	8x	256x256	No	0.451	0.523
7	8x	512x512	Yes	0.451	0.515
8	8x	512x512	No	0.451	0.515

Table 5.5: Accuracy and AUC scores per configuration (averaged over 5 folds)

0.5 is not ideal. There is a slight difference of around 0.06 between the 4x downscale and the 8x downscale. This indicates that the 8x downscale performs slightly better.

When considering the patch size and colour correction, there is no difference. For example, in experiments 3 and 4, the difference lies in the use of colour correction versus no colour correction. However, there is no difference in accuracy or AUC. The same goes when comparing experiments 1 and 3. Here, the only difference is the patch size, but the accuracy and AUC are the same.

5.2.2 Additional Experiments

This section will briefly describe some of the results from additional experiments, as they provided interesting insights. One of these experiments is the one where the data was downsampled by a factor of 20, similar to the data from the segmentation. This experiment used a patch size of 16x16 to compensate for the level of downsampling. It was only run using the colour-corrected dataset. Nonetheless, the results are better than those of the planned experiments. The accuracy is higher by 0.22, and the AUC is higher by at least 0.3. When looking at the con-

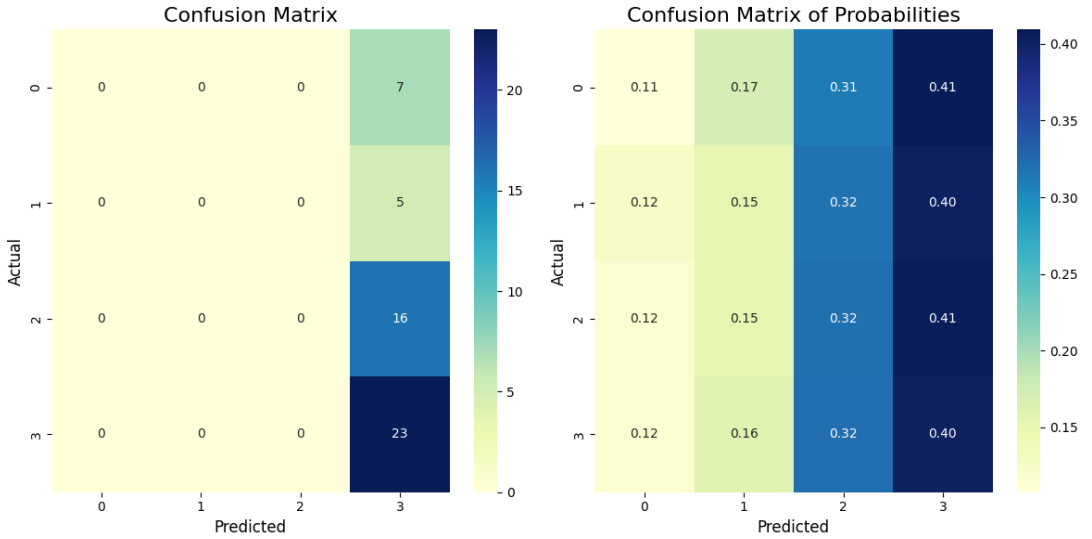

Figure 5.6: The confusion matrix for experiment 6. The confusion matrix of probabilities shows the probabilities for each class.

fusion matrix in Figure 5.7, the model is now able to distinguish the classes from each other. The performance for class 0 is not very good. The model is uncertain about the choice because it has assigned equal probability to both classes 0 and 1 overall. The performance is the worst for class 1, where it has predicted more cases to be class 2 than class 1, even with a higher probability, meaning it is more certain of its mistake. The accuracy and probability increase for class 2. Although it currently predicts many cases as class 2, it is not entirely sure of its decision. When it comes to class 3, both the predictions and the probability are good. This means that it currently identifies the most cases as class 3 and is also very specific about this classification. To conclude, although the results from experiment 9 are not great, they still improve over the planned experiments.

To assess whether the model is unable to classify at higher resolutions or if something else might be the problem, another experiment was conducted at a

Experiment	Downsample	Patch Size	Colour Correction	**Accuracy**	**AUC**
9	20x	16x16	Yes	0.671	0.851
10	16x	16x16	Yes	0.395	0.394
11	16x	16x16	No	0.422	0.392
12	16x	32x32	Yes	0.422	0.474
13	16x	32x32	No	0.439	0.495

Table 5.6: Accuracy and AUC scores per configuration for the unplanned experiments(averaged over 5 folds)

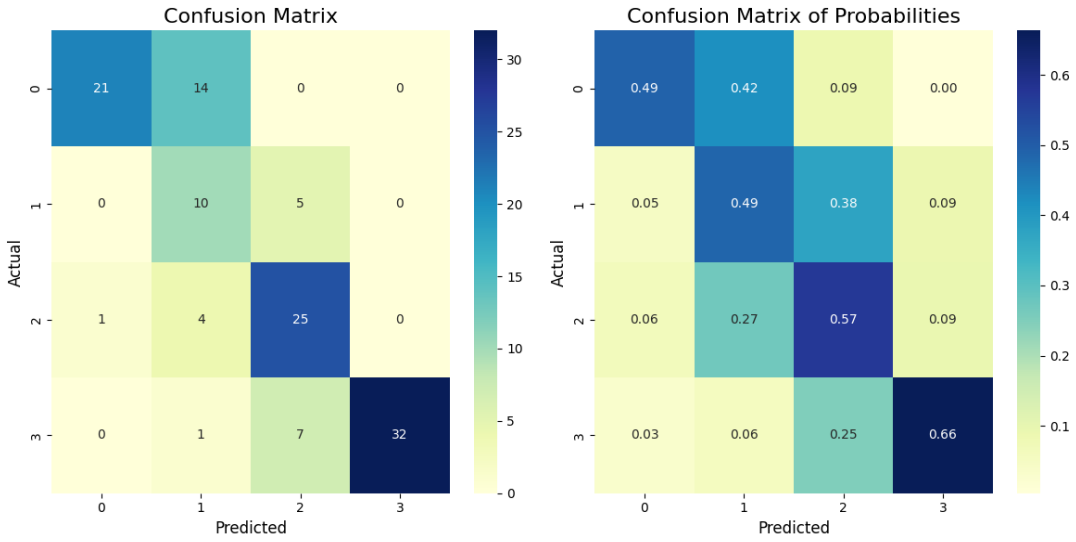

Figure 5.7: The confusion matrix for experiment 9. The confusion matrix of probabilities shows the probabilities for each class.

downsampling rate of 16x with patch sizes of either 16x16 or 32x32. This experiment showed a similar pattern to the results of the 4x and 8x downsampling. The results of these experiments can be found in Table 5.5 at experiments 10-13. Although the experimental setup for 9 was almost the same as that for the downsample and the patch compared to 10-13, there was a notable difference in performance, nearly twice that of the others.

6 Conclusion & Discussion

This research aimed to automate the classification task that experts currently have to perform manually. This task is time-consuming, but more importantly, it lacks consistency across various human experts. It was previously unknown which pre-processing steps and other requirements were necessary to emulate this task. For example, human experts need the colour intensities to be normalised for accurate classification, but is this also necessary for a machine? How well can we segment the grey matter from the image? What classification approach gives an accuracy above a satisfactory level? To answer these questions, we set out to build a pipeline to automate this process. Meanwhile, various experiments were conducted to evaluate the impact of colour correction and to determine the optimal settings for both the segmentation and classification models. In this chapter, we conclude with a summary of our findings and a discussion of their limitations.

6.1 Conclusion

In this research, we have demonstrated the effectiveness of using the nnU-Net framework to segment grey tissue from images. The results presented in section 5.1 show that this model comes very close to the gold standard set out for this task and sometimes outperforms the gold standard. In addition to the model's good performance, we have demonstrated its independence from the label, brain region, tissue area size, and the colour correction. This indicates that the model generalises well and that colour correction is not necessary for segmentation.

All of these facts combined show that nnU-Net can be an effective solution. Additionally, it is very time-saving due to its ease of use. Although time-saving in the development is not a goal, it can be an added benefit.

The planned experiments, along with some additional experiments, suggested that CLAM could not be an effective architecture. Still, one of the additional experiments, the first one to be performed, suggested that it might be possible. This makes it difficult to conclude on the effectiveness of CLAM for this classification problem because further work is required to verify it correctly. Although the results from the segmentation and the classification appear to show no difference in colour correction, the classification results are not reliable enough to justify claiming that the colour correction has no impact.

6.2 Limitations & Future Works

6.2.1 Lack of comparison for segmentation

The goal of the segmentation model was not to get the highest Dice or NSD score. The segmentation needed to be effective enough to be useful for classification. The numbers and visual inspection indicate that it is indeed the case. However, due to the lack of comparison with other network architectures for the segmentation model, we cannot determine how the performance of the segmentation model affects the classification model. Although it may not be the most interesting, it could be investigated whether a different architecture or model behaves differently, thereby improving the classification.

6.2.2 Lack of segmentation evaluation metric

Although Dice is a widely used metric to evaluate performance, it is only as good as the gold standard it compares with. Usually, the gold standard is near-perfect, but in our case, it was not. If the predicted mask is better than the gold standard, the Dice score will decrease instead of increase. Our case could have benefited from a usability metric that tells us how regular the segmentation border is. For example, it could calculate the accumulated length of the segmentation or the irregularity of the border.

Additionally, a downside of the NSD score is that it is not standardised. The reason for this is that the user determines the tolerance. This means it is subject to the user's opinion. There is no standard for the tolerance parameter, neither in research nor in industry.

6.2.3 Performance of the classification model

Unfortunately, it is unclear why the performance of the classification model is so poor. The first experiment performed on the classification framework was the unplanned experiment with the 20x downsampled data. Although theoretically incorrect, it demonstrated that the framework could learn from the patches. This data was generated early on, as it was also necessary for the segmentation. However, subsequent experiments failed to learn from similar data.

The CLAM framework was designed for full-quality images [24], suggesting that downsampling is not necessary. Ashtaiwi et al. [35] indicate that a higher resolution is better for detecting breast cancer. But, Peng et al. [27] found that using a downsampling rate of 8x was the best for detecting glomeruli in Renal Direct Immunofluorescence. Both these examples had comparable resolution to the data in this thesis. Although these domains differ from the case here, the fact remains that the size of the features determines the combination resolution and patch size, and in essence, the model's performance. This is in contrast with the results in this thesis. Combined with the fact that the results yield such a strange outcome, where the model appears to be unable to learn, this suggests that there is something wrong with the code, rather than with the experimental setup. This is further verified by the unplanned experiment with a downsampling rate of 16x. This experiment was conducted most recently to confirm the suspicion that a bug exists in the code. The experiment of 16x downsampling should be similar in this sense to the experiment of 20x downsampling at the beginning, in that it would yield similar results. However, it does not. Its results are analogous to those of the planned experiments, indicating an error somewhere.

6.2.4 Improvements given a successful CLAM implementation

Although it is not proven in this thesis, CLAM may remain an effective architecture for this classification problem. If that turns out to be the case, several hyperparameters and other issues require further investigation.

The first issue to solve is the class imbalance. There are a lot of images for higher class labels than for the lower ones. The results from the 20x downsample show that the model is better at classifying labels 3 and 2. It might be that these are easier to classify, but it could also be that there are more images for these labels to train on.

The second item that requires further research is the encoder used for feature extraction. Currently, the default ResNet encoder is used; however, other encoders may yield better results. Breen et al [36] found that for the classification of ovarian cancer in histopathological images, the UNI foundation model performed better than the ResNet50 encoder. The UNI foundation model is supported as an encoder for the CLAM framework. This foundation model has been pre-trained on over 100 million H&E-stained images. This is more specific to our use case than the general images ResNet has been trained on. Connected with this, further work could be done to investigate what the best dimension size is for the feature vector. The default is currently 1024, however, other sizes may perform better.

The third item that requires further research is the patch size. The issue with having a single, definitive patch size for this specific dataset is that it is challenging to capture every feature. Currently, the amyloid plaques are confined to a single patch, as illustrated in Section 3.2.3. However, the glial cells are approximately

four times smaller than the patch size. Is the model capable of capturing this? Next to this, the large cortical layers, as explained in Section 1.2, are also crucial for the classification. These will not fit a small patch. CLAM addresses this multi-scale challenge through its attention-based multiple instance learning framework, which aggregates information from patches at different spatial locations and scales. However, it is not verified whether CLAM is capable of doing this in this case.

An improvement that can be made in further work is to maintain the same train, validation, and test splits for both the classification and segmentation models. If they are kept the same, we can properly verify how an untrained case performs for both models. Especially since the goal of this research was practical, we want to know how well untrained cases perform. This is an idea we attempted to pursue after completing the training for the segmentation model. However, it turned out that the distribution of the cases for the randomly generated splits for the segmentation was not good and balanced for the classification. This meant that to keep this goal in mind, we had to either retrain the segmentation or accept a performance hit to the classification. The decision was made to forego this idea.

Lastly, not every image at our disposal was used for training and testing due to some technical errors. There were some technical problems, most likely due to the size of the images, that caused the scripts written to upscale the mask, extract the segmentation, create the patches and extract the features to crash. This was the case for about 20 images. Ideally, every image would have been used. However, it was determined that assessing whether CLAM was a usable architecture was more relevant than ensuring that every image was retained during each step. Additionally, the error messages provided were not descriptive, requiring extensive debugging to determine the exact cause of the issue.

6.3 Final thoughts

To conclude, we have examined and proven the effectiveness of nnU-net for this segmentation problem. However, we were unable to create a working classification model using CLAM. We have gained a lot of insights into the problem, but have not yet found the root cause. Nonetheless, the first steps have been made towards an automated classification pipeline that will aid the researchers working with these images.

References

[1] P. Scheltens, B. D. Strooper, M. Kivipelto, *et al.*, "Alzheimer's disease", English, *The Lancet*, vol. 397, no. 10284, pp. 1577–1590, Apr. 2021. DOI: 10.1016/S0140-6736(20)32205-4.

[2] "World Alzheimer Report 2024: Global changes in attitudes to dementia", Sep. 2024. [Online]. Available: https://www.alzint.org/resource/world-alzheimer-report-2024/ (visited on 02/07/2025).

[3] L. Goodman, "ALZHEIMER'S DISEASE: A Clinico-pathologic Analysis of Twenty-three Cases with a Theory on Pathogenesis", *The Journal of Nervous and Mental Disease*, vol. 118, no. 2, p. 97, Aug. 1953, ISSN: 0022-3018.

[4] S. Ayton, Y. Wang, I. Diouf, *et al.*, "Brain iron is associated with accelerated cognitive decline in people with Alzheimer pathology", *Molecular Psychiatry*, vol. 25, no. 11, pp. 2932–2941, Nov. 2020. DOI: 10.1038/s41380-019-0375-7.

[5] S. Ayton, S. Portbury, P. Kalinowski, *et al.*, "Regional brain iron associated with deterioration in Alzheimer's disease: A large cohort study and theoretical significance", *Alzheimer's & Dementia: The Journal of the Alzheimer's*

Association, vol. 17, no. 7, pp. 1244–1256, Jul. 2021. DOI: 10.1002/alz.12282.

[6] S. van Duijn, R. J. Nabuurs, S. G. van Duinen, and R. Natté, "Comparison of Histological Techniques to Visualize Iron in Paraffin-embedded Brain Tissue of Patients with Alzheimer's Disease", *Journal of Histochemistry & Cytochemistry*, vol. 61, no. 11, pp. 785–792, Nov. 2013, Publisher: Journal of Histochemistry & Cytochemistry. DOI: 10.1369/0022155413501325.

[7] F. Cabitza, D. Ciucci, and R. Rasoini, "A Giant with Feet of Clay: On the Validity of the Data that Feed Machine Learning in Medicine", in *Organizing for the Digital World*, F. Cabitza, C. Batini, and M. Magni, Eds., Cham: Springer International Publishing, 2019, pp. 121–136. DOI: 10.1007/978-3-319-90503-7_10.

[8] S. D. Yanowitz and A. M. Bruckstein, "A new method for image segmentation", *Computer Vision, Graphics, and Image Processing*, vol. 46, no. 1, pp. 82–95, Apr. 1989. DOI: 10.1016/S0734-189X(89)80017-9.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", en, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Cham: Springer International Publishing, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[10] R. Azad, E. K. Aghdam, A. Rauland, *et al.*, "Medical Image Segmentation Review: The Success of U-Net", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10 076–10 095, Dec. 2024, Confer-

ence Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. DOI: 10.1109/TPAMI.2024.3435571.

[11] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation", *Nature Methods*, vol. 18, no. 2, pp. 203–211, Feb. 2021, Publisher: Nature Publishing Group. DOI: 10.1038/s41592-020-01008-z.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, arXiv:2010.11929 [cs], Jun. 2021. DOI: 10.48550/arXiv.2010.11929.

[13] A. Hatamizadeh, Y. Tang, V. Nath, *et al.*, "UNETR: Transformers for 3D Medical Image Segmentation", in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2022, pp. 1748–1758. DOI: 10.1109/WACV51458.2022.00181.

[14] H. Cao, Y. Wang, J. Chen, *et al.*, "Swin-Unet: Unet-Like Pure Transformer for Medical Image Segmentation", en, in *Computer Vision – ECCV 2022 Workshops*, L. Karlinsky, T. Michaeli, and K. Nishino, Eds., Cham: Springer Nature Switzerland, 2023, pp. 205–218. DOI: 10.1007/978-3-031-25066-8_9.

[15] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu, "Swin UNETR: Swin Transformers for Semantic Segmentation of Brain Tumors in MRI Images", in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, A. Crimi and S. Bakas, Eds., Cham: Springer International Publishing, 2022, pp. 272–284. DOI: 10.1007/978-3-031-08999-2_22.

[16] A. Kirillov, E. Mintun, N. Ravi, *et al.*, "Segment Anything", in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Oct. 2023, pp. 3992–4003. DOI: 10.1109/ICCV51070.2023.00371.

[17] D. Komura, M. Ochi, and S. Ishikawa, "Machine learning methods for histopathological image analysis: Updates in 2024", en, *Computational and Structural Biotechnology Journal*, vol. 27, pp. 383–400, 2025. DOI: 10.1016/j.csbj.2024.12.033.

[18] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, ISSN: 0302-9743, 1611-3349, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 411–418. DOI: 10.1007/978-3-642-40763-5_51.

[19] B. Ehteshami Bejnordi, M. Veta, P. Johannes van Diest, *et al.*, "Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer", *JAMA*, vol. 318, no. 22, pp. 2199–2210, Dec. 2017. DOI: 10.1001/jama.2017.14585.

[20] N. Coudray, P. S. Ocampo, T. Sakellaropoulos, *et al.*, "Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning", *Nature Medicine*, vol. 24, no. 10, pp. 1559–1567, Oct. 2018, Publisher: Nature Publishing Group. DOI: 10.1038/s41591-018-0177-5.

[21] Y. Fu, A. W. Jung, R. V. Torne, *et al.*, "Pan-cancer computational histopathology reveals mutations, tumor composition and prognosis", *Nature Cancer*,

vol. 1, no. 8, pp. 800–810, Aug. 2020, Publisher: Nature Publishing Group. DOI: 10.1038/s43018-020-0085-8.

[22] G. Campanella, M. G. Hanna, L. Geneslaw, *et al.*, "Clinical-grade computational pathology using weakly supervised deep learning on whole slide images", *Nature Medicine*, vol. 25, no. 8, pp. 1301–1309, Aug. 2019, Publisher: Nature Publishing Group. DOI: 10.1038/s41591-019-0508-1.

[23] M. Ilse, J. Tomczak, and M. Welling, "Attention-based Deep Multiple Instance Learning", in *Proceedings of the 35th International Conference on Machine Learning*, ISSN: 2640-3498, PMLR, Jul. 2018, pp. 2127–2136.

[24] M. Y. Lu, D. F. K. Williamson, T. Y. Chen, R. J. Chen, M. Barbieri, and F. Mahmood, "Data-efficient and weakly supervised computational pathology on whole-slide images", *Nature Biomedical Engineering*, vol. 5, no. 6, pp. 555–570, Jun. 2021, Publisher: Nature Publishing Group. DOI: 10.1038/s41551-020-00682-w.

[25] D. A. Bennett, A. S. Buchman, P. A. Boyle, L. L. Barnes, R. S. Wilson, and J. A. Schneider, "Religious Orders Study and Rush Memory and Aging Project", *Journal of Alzheimer's disease: JAD*, vol. 64, no. s1, S161–S189, 2018. DOI: 10.3233/JAD-179939.

[26] Netherlands Brain Bank, *About the Netherlands Brain Bank*. [Online]. Available: https://www.brainbank.nl/about-us/ (visited on 04/07/2025).

[27] C. Peng, K. Zhao, A. Wiliem, *et al.*, "To What Extent Does Downsampling, Compression, and Data Scarcity Impact Renal Image Analysis?", in *2019 Digital Image Computing: Techniques and Applications (DICTA)*, Dec. 2019, pp. 1–8. DOI: 10.1109/DICTA47822.2019.8945813.

[28] M. Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, *Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation*, arXiv:1802.06955 [cs], May 2018. DOI: 10.48550/arXiv.1802.06955.

[29] F. Isensee, T. Wald, C. Ulrich, *et al.*, "nnU-Net Revisited: A Call for Rigorous Validation in 3D Medical Image Segmentation", in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, M. G. Linguraru, Q. Dou, A. Feragen, *et al.*, Eds., Cham: Springer Nature Switzerland, 2024, pp. 488–498. DOI: 10.1007/978-3-031-72114-4_47.

[30] T. Sørensen, *A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons* (Kongelige Danske Videnskabernes Selskab). Munksgaard, 1948.

[31] L. R. Dice, "Measures of the Amount of Ecologic Association Between Species", en, *Ecology*, vol. 26, no. 3, pp. 297–302, Jul. 1945. DOI: 10.2307/1932409.

[32] V. Raina, N. Molchanova, M. Graziani, *et al.*, "Tackling Bias in the Dice Similarity Coefficient: Introducing NDSC for White Matter Lesion Segmentation", in *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, Apr. 2023, pp. 1–5. DOI: 10.1109/ISBI53787.2023.10230755.

[33] L. Maier-Hein, A. Reinke, P. Godau, *et al.*, "Metrics reloaded: Recommendations for image analysis validation", *Nature Methods*, vol. 21, no. 2, pp. 195–212, Feb. 2024, Publisher: Nature Publishing Group. DOI: 10.1038/s41592-023-02151-z.

[34] A. Reinke, M. D. Tizabi, C. H. Sudre, *et al.*, *Common Limitations of Image Processing Metrics: A Picture Story*, Version Number: 8, 2021. DOI: 10.48550/ARXIV.2104.05642.

[35] A. Ashtaiwi, "Optimal Histopathological Magnification Factors for Deep Learning-Based Breast Cancer Prediction", *ResearchGate*, Mar. 2025. DOI: 10.3390/asi5050087.

[36] J. Breen, K. Allen, K. Zucker, L. Godson, N. M. Orsi, and N. Ravikumar, "A comprehensive evaluation of histopathology foundation models for ovarian cancer subtype classification", en, *npj Precision Oncology*, vol. 9, no. 1, p. 33, Jan. 2025, Publisher: Nature Publishing Group. DOI: 10.1038/s41698-025-00799-8.

Appendix A Figures

A.1 Dice and NSD Scores

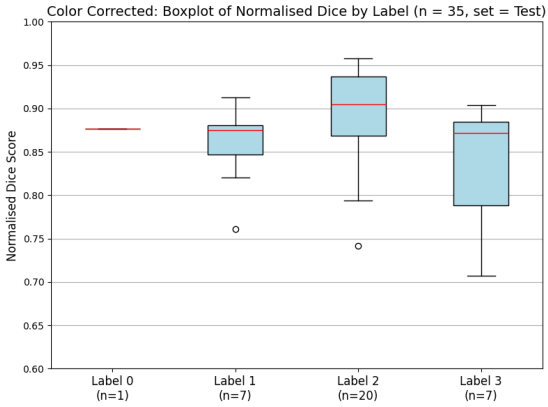

Figure A.1: An overview of the Dice scores for the colour corrected dataset split out by label

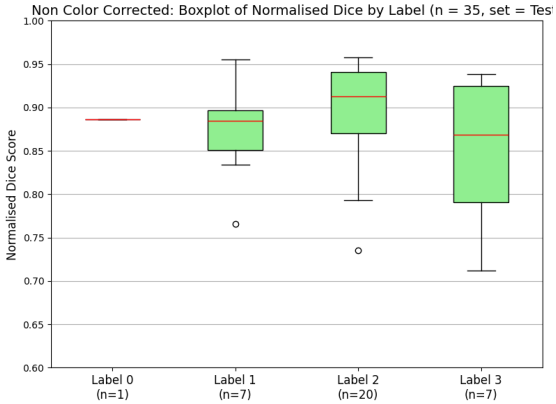

Figure A.2: An overview of the Dice scores for the non-colour corrected dataset split out by label

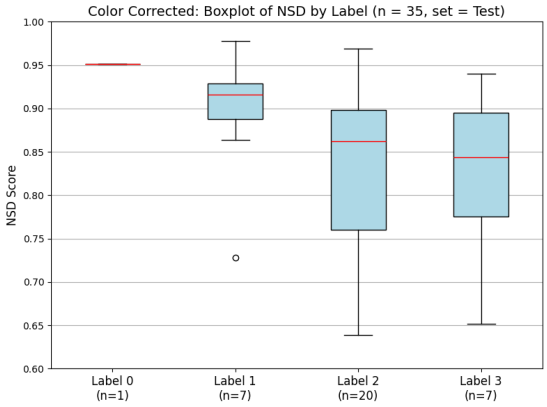

Figure A.3: An overview of the NSD scores for the colour corrected dataset split out by label

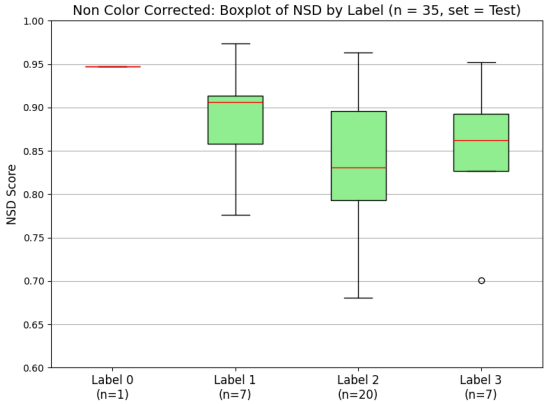

Figure A.4: An overview of the NSD scores for the non-colour corrected dataset split out by label

Appendix B Usage of AI

Code Generation

I have utilised GitHub CoPilot and ChatGPT for certain aspects of code generation. These parts were used to aid in programming the scripts needed for visualisation. They did not alter the data in any way and only saved time by writing the required function for visualising the data. I have reviewed all the results to confirm their accuracy.

Grammar Checking

I have Grammarly Pro for spell checking. This ranged from ensuring that the English grammar was correct to rewriting certain parts of the sentence for greater clarity. It also changed some words to reduce monotony and increase variability in word usage.

Rewriting for clarity

I have used Claude AI to rewrite specific paragraphs for clarity. When I had written a paragraph containing my personal ideas and was not satisfied with the way I had written it, I did not entirely know how to improve it. I then used Claude AI to rewrite the part. By writing this piece myself first, the ideas are still mine, and I retain complete control over the content of the paragraph. This way it is better readable for other audiences.