University of Copenhagen

Computer Science

Master's Thesis

Colitis Treatment Predictions Using Machine Learning and Endoscopy Videos

Author:
Casper Bresdahl

Supervisor:
Bulat Ibragimov

May 30, 2022

Contents

1 Abstract

In this thesis it was investigated how machine learning could be used in the treatment and diagnosis of colitis. Colitis is a chronic digestive disease and is characterised by inflammation in the colon. This characteristic was exploited and it was first investigated how well a 2D ResNet model could classify individual frames from an endoscopy examination. The best performing model achieved 65% accuracy, although visual inspection showed the model was overfitting the training data and overly predicting inflammation. The resulting segmentation of the endoscopy videos were then used to train a 1D ResNet model predicting which of five treatments the patient had received. When trained only on predicted data the model achieved 40% accuracy, and when adding the artificially constructed true segmentations to the training data, the model achieved 48% accuracy. It was then attempted to train a 1D U-Net model to refine the segmentations and bring more structure to them. By visual inspection this was to some extent achieved although overly many frames were overturned to be healthy. Lastly, using these refined segmentation results, a new 1D ResNet model was trained to again predict the treatment received by patients. This time achieving 50% accuracy when only trained on predicted data and 20% accuracy when the true segmentations were added.

2 Introduction

In this thesis project we will look into how machine learning can be used in the treatment of colitis. Colitis is a chronic digestive disease which is characterized by inflammation of the inner lining of the colon. Suffering from colitis causes pain and discomfort to the abdomen and often causes diarrhea with or without blood[1]. Often four different treatments are prescribed, depending on the scope of inflammation in the colon. Local 5-ASA is a mild suppository if the inflammation does not stretch too far into the colon. Oral 5-ASA is a mild treatment and is prescribed if the inflammation is mild but stretches far into the colon. Oral steroid is a moderate treatment and is prescribed if the inflammation is severe, but not necessarily stretched far into the colon. Lastly IV steroids is a harsh treatment prescribed if the inflammation is severe and is stretched far into the colon.

In this thesis work, several deep neural networks were trained to detect inflammation in individual frames of endoscopy examinations in an attempt to draw segmentations of these videos. As the inflammation stretches into the colon without 'holes', it was attempted to predict this separation point, where we go from an inflamed colon to a healthy colon. Such a segmentation and separation point could then be used by a doctor to determine the correct treatment. However, the segmentations were also used to train other machine learning models to predict which treatment patients should be prescribed. One could imagine these results being used as a second opinion for a doctor.

The thesis work is divided into five sections. In the first, the theory used for this project is presented. In the second section the methods used throughout the thesis work is presented and goes into details about data pre-processing, how separation point predictions were made, how treatment predictions were made and how the predictions were refined and then used for predictions again. Next the results of the thesis work will be presented before we go into a discussion about the results and end with a conclusion.

3 Theory

In this section the theory used throughout the thesis work is presented. This includes a presentation of the original model architectures used, the loss functions which were used, the optimization algorithm the models were trained with, various regularization techniques used to avoid overfitting and how to interpret ROC curves.

3.1 Deep neural networks

Deep neural networks is a branch of machine learning characterized by having several layers between the input and the output. Deep neural networks do not require manual engineering as they are capable of automatic discovery of tendencies and patterns in the training data, without any prior knowledge of the data. Deep neural networks employ the idea that natural high-level signals very often are composed of lower level signals, and thus by composing many layers of neurons in succession forms a deep structure which is capable of learning features on multiple levels of abstraction [2].

Common for all deep neural networks is, they are composed of the same basic components: neurons, weights, biases and activation functions. Each layer holds a number of neurons where each has its own weights and bias associated with it. The neurons in one layer is connected to a subset or all of the neurons in the next layer, feeding the next layer in the network with inputs. A neuron, its weights and its bias constitutes a linear function, but for the network to adapt to complex tendencies in the data, a non-linear activation function is applied to the output of the neuron. Common activation functions are *ReLu*, *Softmax* and *sigmoid*.

During training a loss function is formulated and an optimization strategy employed to tune the parameters of the model optimally. Common loss functions are *mean squared error* and *cross entropy* and common optimization strategies include *stochastic gradient descent* and *Adam*. The loss function is used to estimate how well the model predicted, and effectively measures how much the model needs to be corrected to perform optimally. The optimization strategy is how the model should be changed according to the measured loss.

3.2 Convolutional neural networks

One type of deep neural networks are convolutional neural networks which are primarily build from convolutions. This allows the networks to work on images as a convolutional layer learns the weights of its convolution kernels. Convolutional neural networks are most commonly used for segmentation or classification tasks which requires processing of images [3].

3.2.1 U-Net

U-Net is a convolutional neural network architecture used for segmentation tasks. The network can be split in two parts, one which downscales the image, and one which upscales the image afterwards. The first part consists of three blocks where, in each, two convolutions are made before a max-pooling is performed. During the first convolution in a block the number of feature channels is doubled, and during the max-pooling the images are roughly halved in size. As the kernel sizes stays constant, and thus covers a larger portion of the images, this allows the U-Net to search for higher level features in the images. The second part also consists of three blocks with two convolutions each, however, now the number of feature channels are halved during the first convolution of the block, and the images are growing to twice its size after each upsampling is performed. After each upsampling a concatenation from the corresponding left part of the network is performed. These

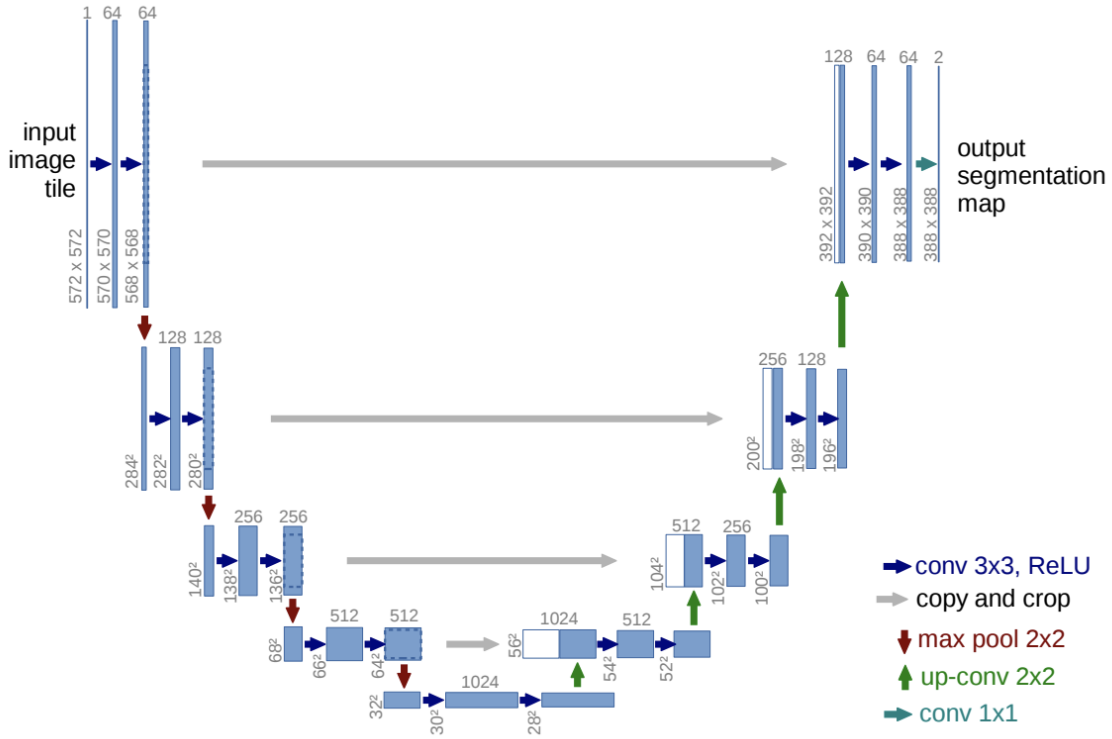


Figure 1: U-Net model architecture for an input image of size 572x572. The number of feature channels is denoted on top of each blue box, and the image size is denoted on the side of the blue boxes. Illustration taken from [4].

skip connections allows the model to recover lost spatial information and alleviates vanishing gradient issues. In the end, a 1x1 convolution is performed to map the output into the correct number of prediction classes. The output of U-Net is a segmentation of the input, and thus has the same size as its input image. An example of a U-Net can be seen in Figure 1 [4].

3.2.2 ResNet

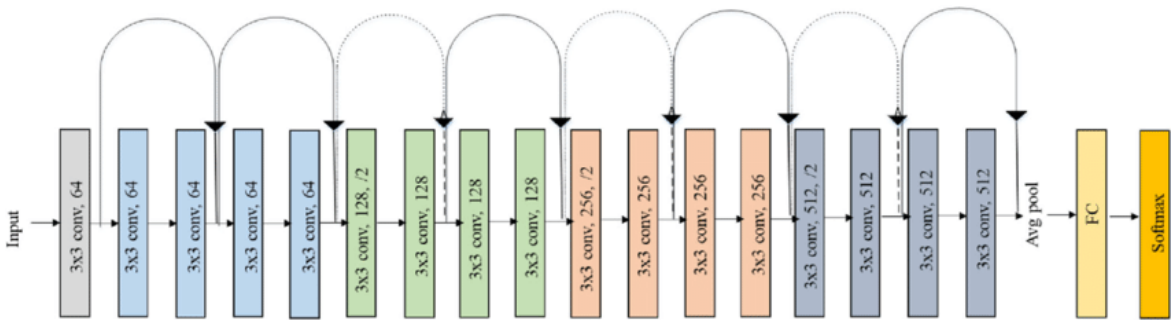


Figure 2: ResNet-18 model architecture consisting of a main path and identity residual connections. Illustration is taken from [7].

ResNet is a convolutional neural network used for classification. The model comes in several sizes, but the core building blocks are the same across all variants. The ResNet consists of a main path and residual shortcuts. Looking at ResNet-18, the main path begins with a 7x7 convolution, a 3x3 max-pooling followed by four blocks, with four convolutions in each. At the end a global average-pooling is performed followed by

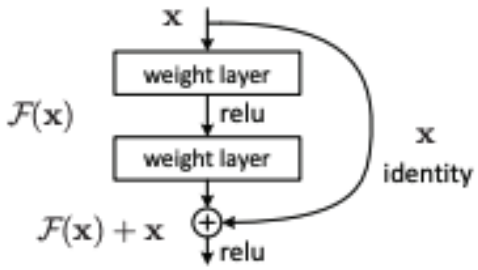


Figure 3: Illustration of how the residual connections skip parts of the ResNet model. Illustration taken from [5].

a fully connected layer with a softmax. The residual shortcuts allows for some of the data to skip ahead in the network as an identity from previously in the network and is added to the output of a later convolution. These shortcuts can be directly used when the images are of same size, i.e. in the same block, but if a skip leads to a new block, padding with zeros or performing a 1x1 convolution is needed. The residual connections helps to alleviate vanishing gradients. The ResNet-18 architecture can be seen in Figure 2, and an illustration of how the residual connection works can be seen in Figure 3 [5].

As mentioned does ResNet come in several sizes where the number of blocks in the main path remain the same, but the number of convolutions in each varies from four in ResNet-18 to 108 in the block with most convolutions in ResNet-152. This makes a large difference in both depth of the network but also in the number of parameters the models contain. This makes it more difficult to train the larger versions, but it also makes the models more powerful as both the top-1 error and the top-5 error gets lower the more parameters the ResNet have [6].

3.3 Loss functions

3.3.1 Cross entropy

Cross entropy is a loss function which builds on the concept of entropy, and measures the difference between two probability distributions for a set of events, by computing the total entropy between the two probability distributions. The entropy is the number of bits requited to transmit a randomly selected event from a probability distribution. An asymmetric distribution will have low entropy due to the larger probability of certain events, whereas a more uniform distribution will have a larger entropy due to the events having closer to equal probability. The cross entropy is then the average number of bits required to encode an event from source distribution P when using model distribution Q. This can be seen as the number of additional bits needed to represent an event using an approximating distribution Q rather than the original distribution P [8].

In terms of a neural network and a multi class classification problem, only one class is considered correct, thus we can one-hot encode the target vector $\mathbf{y}$, and use the result of a softmax activation as a prediction vector $\hat{\mathbf{y}}$. We can then compute the average cross entropy as:

$$CE = \frac{1}{N}\left(\sum_{i=1}^{N} \mathbf{y}_i \cdot log(\hat{\mathbf{y}}_i)\right)$$

For N being the size of the training set.

Binary cross entropy is a special case of cross entropy where we only have two classes. This means the target

y is now either 1 or 0, and $\hat{y}$ is the model probability. The average binary cross entropy can be defined as:

$$BCE = \frac{1}{N}\sum_{i=1}^{N}[y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i)]$$

For *N* being the size of the training set. Cross entropy loss and binary cross entropy loss are commonly used for both segmentation and classification tasks.

3.3.2 F1-score

DICE coefficient or F1-score is a metric combining precision and recall. It can be defined as:

$$F1 = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Where TP is the number of true positive predictions, FP is the number of false positive predictions and FN is the number of false negative predictions.
In terms of a binary segmentation task, it can also be formulated as two times the intersection between the target and predictions divided by the union of those:

$$F1 = \frac{2 \cdot y \cap \hat{y}}{y \cup \hat{y}}$$

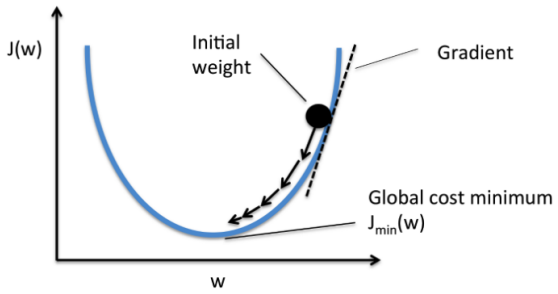
Where y is the target and $\hat{y}$ is the predictions. This can intuitively be thought of as two times the number of correctly predicted pixels belonging to the segmentation class divided by the sum of the true number of pixels belonging to the segmentation class and the predicted number of pixels belonging to the segmentation class. Intuitively, if the models are accurate, the numerator will be half of the denominator, hence the multiplication by two. To convert the score to loss, we need to have $1 -$ F1-score.
The advantage of evaluating model performance based on F1-score comes when the segmentation class is small compared to the total number of pixels in the image. Often it is easy to predict the background correctly, and thus, in tasks where large parts of the image is considered background, the accuracy of the model will inherently become large. However, this can be very misleading if very few, or any, predictions are correct in accordance to the segmentation class. Here the F1-score will be a much more accurate measure of model performance as it does not perform a pixel based accuracy measure, but measures the overlap of correct predictions belonging to the segmentation class.

3.4 Optimization

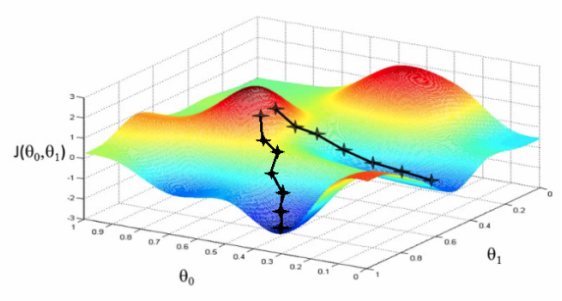
3.4.1 Gradient descent

Gradient descent is a first-order iterative optimization algorithm used to find a minimum of a differentiable function, and is widely used because of its simplicity and relative cheap computation cost. Gradient descent is based on the observation that if some function is differentiable in a neighbourhood, then its function value decreases fastest if one takes a small step in the direction of the negative gradient of the function. If small enough steps are taken in an iterative fashion, then it can be guaranteed that the algorithm converges to a local minima. When the function is convex, all local minima are global minima, and in this case the algorithm is guaranteed to converge to a global solution. More concretely, an initial guess, $\mathbf{x}_0$, for a local minima is made, and from here the position is iteratively updated as $\mathbf{x_{n+1}} = \mathbf{x_n} - \gamma\nabla\mathbf{F}(\mathbf{x_n})$, where *F* is the function, and γ is the step size. Intuitively the algorithm can be thought of as having a ball rolling on a hill landscape where hill tops would correspond to local optima and valley bottoms would correspond to local minima. Placing the ball somewhere in this landscape will have it roll towards and stop near a valley bottom. As gradient descent only uses local information and simply takes a step in the steepest descent direction, it is prone to get stuck in local minima when used on pathological functions. If the step size is kept constant or decaying with a fixed

rate, gradient descent can be slow at converging if the functions is very flat, and it can easily overshoot a minima if the function is very 'peaky' [9]. 1D and 2D examples of gradient descent can be seen in Figure 4.



(a) Example of gradient descent in 1D, where the ball slowly rolls towards the global minimum.



(b) Example of gradient descent in 2D, where two different initializations results in the algorithm terminating in two different minima.

Figure 4: Examples of gradient descent. Illustrations taken from [12][11].

3.4.2 Adam

Adaptive moment estimation, Adam, is a stochastic gradient descent based optimization algorithm which is commonly used in all kinds of machine learning models. Adam makes three important improvements to traditional gradient descent. The first being each parameter in the model has its own adaptive learning rate. This means the step sizes for adjusting each parameter in the model will dynamically and independently be adjusted throughout training. The second improvement is, each learning rate can obtain momentum. This is important because as previously described, gradient descent can be slow at converging if the function is very flat, or it can overshoot a minima if the function is 'peaky'. With momentum, the prior gradient steps are taken into consideration when a new gradient step is performed, as a weighted average of gradients are computed. This results in less oscillation towards the minima, the option to get out of a local minima if large steps were taken but a flat spot is suddenly hit and the option to not immediately accelerate if small steps were taken but suddenly falling over a cliff. The last improvement is that, in general the learning rates should be larger at the beginning of training where we usually would be far away from a minima, and smaller towards the end where we need to be careful not to overshoot the minima [10].

3.5 Regularization

Regularization is commonly used in machine learning to close the gap between performance on known and unknown data. As the end goal of machine learning is for the model to perform optimally on all data, including rare cases and unknown data, we want the model to generalize and learn a wide variety of features in the training data. However, if the model can simply look at a single common feature appearing in the training data, the model might weight this particular feature very heavily, and when it is then presented to unknown data where this feature is not present, the performance significantly drops. For this reason, regularization is employed to ensure the model learns a variety of features and is not reliant on observing one in particular. A wide range of regularization techniques exists, in the following four will be introduced.

3.5.1 Dropout

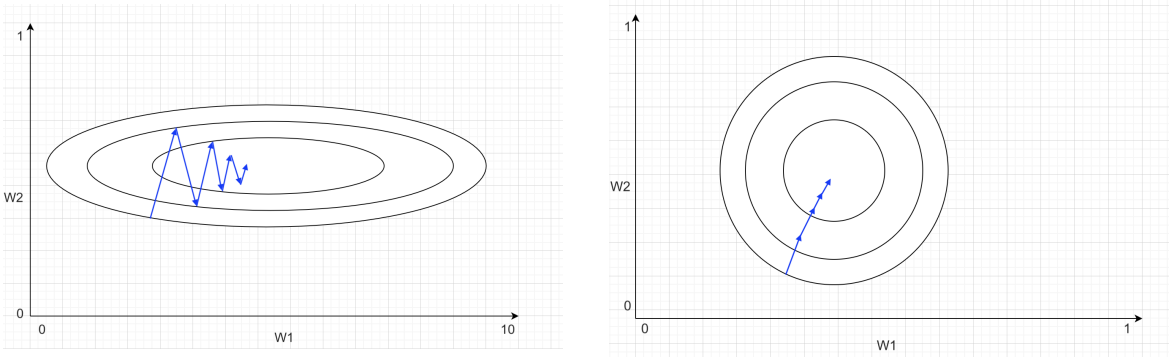
Dropout is a regularization technique where a neuron with some probability p is 'dropped' from the network. This means the neuron along its incoming and outgoing connections are temporarily removed from the network. At each epoch a new set of 'dropped' neurons are chosen according to the dropout probability p, and the remaining neurons are thus forced to learn how to work with a randomly chosen set of other neurons. This drives each neuron towards creating useful features on its own, without relying on other

neurons later correcting its mistakes. Dropout can be interpreted as adding noise to the neurons which forces the optimization to adopt random noise and thus generalize better. However, due to the added noise, the gradients computed during optimization are less accurate and the total number of training epochs needs to be increased [13].
Dropout is only employed during training as to gain the full potential of the network during testing. When we activate all neurons, the expected value of the next layer in the network increases by a factor of $\frac{1}{1-p}$ which leads to incorrect predictions. To avoid this we scale the weights by $\frac{1}{1-p}$ during training, or we can scale the weights by p during testing [13].

3.5.2 Weight decay

A simple solution to overfitting would be to decrease the number of parameters in our model. This would heavily limit its ability to overfit the training data, however, it would also heavily limit its ability to do predictions. More parameters can roughly be translated to more interactions between various parts of the network, and more interactions mean more non-linearities which helps us solve complex problems. However, the complexity of these interactions and of the model can get out of hand. To alleviate this, the squared sum of all weights in the model might be added as a term to our loss function to penalize complex models. This might, however, result in the loss getting so large, that the optimal weights are simply zero. For this reason, we introduce the weight decay as a small constant which is multiplied by the squared sum of the weights such that this regularization term is heavily reduced, and it is no longer optimal to have all weights being zero [14].

3.5.3 Normalization



(a) Non standardized data

(b) Standardized data

Figure 5: Examples of gradient descent optimization on non-standardized data and on standardized data. The blue line is the gradient steps taken during optimization. If the data is not standardized, the error surface becomes elongated, and the gradient steps points away from the minima, resulting in the gradient steps becoming 'zig-zaggy' and inefficient. If the data is standardized, the error surface becomes less elongated, and the gradient steps points more directly towards the minima, resulting in more efficient gradient steps.

Often deep neural networks trained with gradient based optimization algorithms suffer from slow learning when trained on data that is not standardized, i.e. does not have zero mean and unit variance [15]. This should be seen as if the data is not standardized, the error surfaces gets elongated and as a result the gradient step might point away from the minimum. In Figure 5a we see a training example of not having standardized data. The blue line represent the gradient steps taken, which are perpendicular to the contour lines giving the steepest descent direction. Because the data is not standardized, learning becomes 'zig-zaggy'. If we however standardize the data, the error surface is less elongated and the steepest descent direction points more directly towards the minima, giving us the example in Figure 5b, where the gradient steps are much more efficient and learning is thus much faster [16].

Because of this, when working with images, *intensity normalization* is often employed. This implies each colour channel is standardized. In a medical setting, where several different imaging machines may have been used to obtain the data it can also help to standardize the data, as each machine might have slightly different outputs on the same input.
It is not only during the beginning of training we can benefit from normalization. Because the neuron's weights change during training, it is likely the distribution of the data between layers in the network changes, which is referred to as an internal covariate shift [17]. Internal covariate shifts can be reduced by introducing normalization into the network in the form of normalization layers. One such layer is known as a *batch normalization* layer. In batch normalization the mean and variance is computed over each mini batch for a given layer:

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i, \quad \sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2$$

For m being the number of samples in the mini batch. This mean and variance is then used to standardize the input:

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

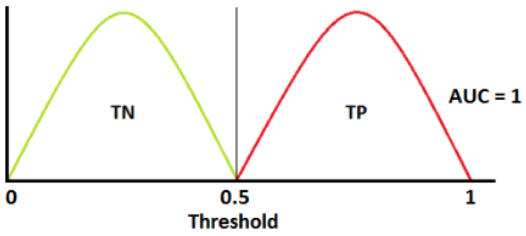
Where ϵ is introduced for numerical stability. Learnable parameters γ and β are then introduced to scale and shift the standardized data again:

$$y_i = \gamma\hat{x}_i + \beta$$

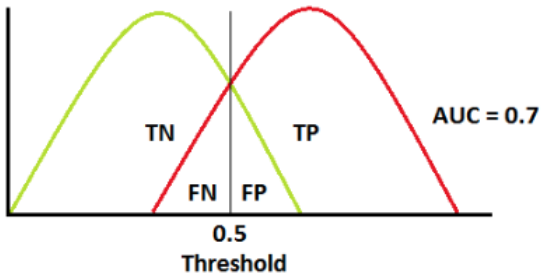
This is done because standardizing the data might constraint what a layer in the network is able to represent. An example would be that normalizing the the inputs of a sigmoid would constrain them to the linear regime of the non-linearity [17]. Thus, simply setting $\gamma = \sqrt{\sigma^2}$ and $\beta = \mu$ allows us to recover the un-normalized data, if that is optimal [16][17].

3.6 ROC curve

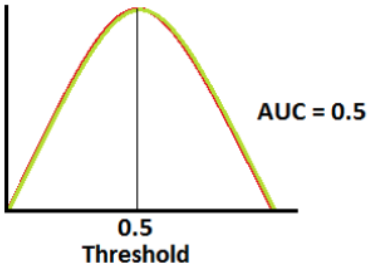
When training binary classifiers a common performance measure is to draw the receiver operating characteristics (ROC) curve. The curve is drawn by plotting the true positive rate against the false positive rate for several thresholds. The true positive rate is computed as $\frac{TP}{TP+FN}$ where *TP* is the true positives, and *FN* is the false negatives. The false positive rate is computed as $\frac{FP}{TN+FP}$ where *FP* is the false positives and *TN* is the true negatives. In a binary classification problem the class with a predicted probability of more than 0.5, is usually the class one denotes as the class the model predicted. But the threshold of 0.5 could arbitrarily be changed, and by changing it we can draw ROC curves [18]. ROC curves can thus be read as a measure of how many more false positives we introduce if we wish to increase the number of true positives.
The evaluation of a ROC curve is usually done by measuring the area under the curve which is between 0 and 1 with a higher area under the curve being better. If we think of the true negatives and the true positives as two distributions, we achieve the best results when the two distributions do not overlap. In this case we get a perfect separability of the two cases, and we get an area under the curve of 1. If the two distributions begins to overlap, however, we begin to see false negative and false positive predictions, and we get a slightly lower area under the curve. In the worst case, the two distributions match and lies on top of each other, and there is no distinction between true positives and true negatives. Here we get an area under the curve of 0.5. When the model predicts all class 1 samples as class 0, and vice versa, we see an area under the curve of 0. These scenarios are illustrated in Figure 6 [18].



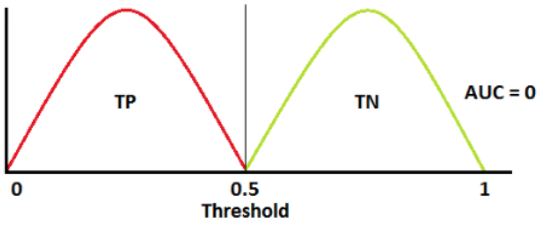
(a) When the two distributions do not overlap.



(b) When the two distributions slightly overlap.



(c) When the two distributions are in indistinguishable.



(d) When the model predicts all samples wrong.

Figure 6: Examples of how different area under the curve are achieved. Illustrations are from [18].

4 Methods

This section will go into details about the methods of the thesis work. We will go into details about how the data is preprocessed, how the different datasets are constructed, details about the main contributions of the thesis work and which model architectures was used.

For the remaining of the thesis work a naming convention has been adopted to make it easier to reference models, datasets and videos. Each video has been enumerated and will be referenced with an index between 0 and 35. Each constructed dataset will include the indices of the videos which are used to compose the datasets along with the number of frames *skipped* before a frame from the video is sampled to the dataset. That is, not every frame from a video is used, only every *x'th* frame is sampled. The models trained will for convenience share name with the dataset it was trained on.

4.1 Data pre-processing

The data used in this project comes from a collaboration partner at Hvidovre Hospital and consists of 47 videos of endoscopy examinations with the purpose of planing a treatment for colitis. Out of these, 35 was found suitable for separation point predictions and 30 was found suitable for treatment predictions. The main reasons for some videos not being suitable are missing treatment annotation, multiple transitions between healthy and inflamed tissue and some videos being filmed outside the colon. The length of the videos vary from 56 to 4989 frames, and is a mix of all frames being healthy/inflamed, biased towards healthy/inflamed or being fairly balanced. The separation points was annotated as a timestamp and was converted to a frame number by computing the frame rate of each recording and then multiplying this with the timestamp of the annotated separation point. Each frame's original size was 1072 by 1920 pixels and was cropped to only contain the actual footage, intensity normalized and resized to 128 by 165 pixels to speed up training while preserving aspect ratio.

4.2 Separation point predictions

For this task, the objective is to find the transition point where the inflammation ends and the colon tissue becomes healthy. For this, we perform a binary classification on each frame in a video to obtain a segmentation of that video. In the ideal case this would show a continuous block of inflammation predictions followed by a continuous block of healthy predictions. The obtained segmentation would then be used in a post-processing step to obtain the separation points.
Several models were trained on different compositions of datasets to evaluate the performance of different factors. In total eight datasets were constructed, each composed for different reasons:

1. Dataset **Idx_4_skip_10** is composed of 499 frames from a single video containing roughly the same number of healthy and inflamed frames. Every 10th frame from the video was sampled to keep the size down, and to avoid overfitting by having too many too similar frames. This dataset was made to evaluate the performance of training on a single video.

2. Dataset **Idx_4_5_skip_20** is composed of 500 frames from two videos from the same patient containing roughly the same number of healthy and inflamed frames. Every 20th frame from the videos was sampled. This dataset was composed to evaluate the performance of using a limited number of videos from the same patient.

3. Dataset **Idx_2_3_4_5_6_skip_50** is composed of 491 frames from five videos of the same patient, which is all the videos of that patient. This dataset has a slight bias towards inflamed data. Every 50th frame was sampled. This dataset was composed to evaluate the performance of using all the available data from a single patient while limiting the size of the training data to avoid overfitting.

4. Dataset **Idx_2_3_4_5_6_skip_20** is composed of 1226 frames from the same videos as the previous dataset, but sampling every 20th frame. This dataset was composed to evaluate the effect of increasing the number of training data in relation to the previous dataset.
5. Dataset **Idx_4_14_18_20_32_skip_20** is composed of 997 frames from five different videos, each from a different patient. Every 20th frame was sampled from the videos, and it is slightly biased towards healthy data. This dataset was composed to evaluate the performance of training on images from different patients while limiting the number of training data to avoid overfitting.
6. Dataset **Idx_4_14_18_20_32_skip_5** is composed of 3978 frames from the same videos as the previous dataset, but every fifth frame was sampled. This dataset was constructed to evaluate the effect of increasing the number of training data in relation to the previous dataset, and to evaluate the performance of a large dataset.
7. Dataset **Idx_3_23_skip_10** is composed of 829 frames from two videos from different patients. Every 10th frame was sampled. This dataset is heavily biased towards inflamed data, as 92% of the data is categorized as inflamed. This dataset was constructed to evaluate the performance of a dataset heavily biased towards inflamed data.
8. Dataset **Idx_19_24_skip_5** is composed of 600 frames from two videos of different patients. Every fifth frame was sampled. This dataset is heavily biased towards healthy data, as 86% of the data is categorized as healthy. This dataset was constructed to evaluate the performance of a dataset heavily biased towards healthy data.

To do the classification a slightly modified 2D ResNet model was used. The modification lies in four fully connected layers and four dropout layers added after the original fully connected layer of size 1000. The added layers allows for a smooth transition towards two classes by approximately halving the features size at each added layer. The dropout layers are added to avoid overfitting, and to increase the model robustness and generalization. The model architecture can be seen in Figure 7.

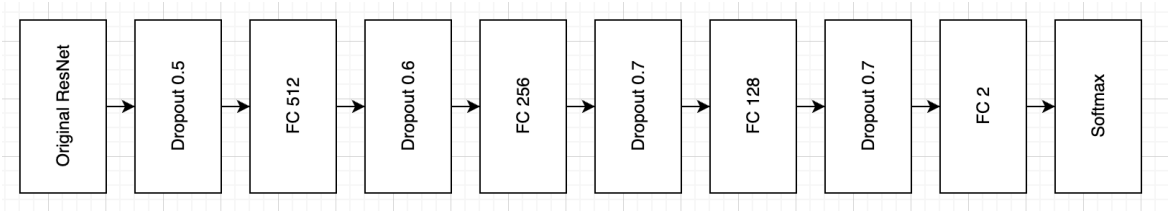


Figure 7: Modified 2D ResNet model used for separation point predictions. The modifications lies in the added fully connected layers after the original fully connected layer of size 1000, and the added dropout layers.

4.3 Post-processing

After using the 2D ResNet model to get segmentation predictions of each endoscopy video, the following post-processing approaches were used to translate them into concrete separation predictions.

4.3.1 Random Walker approach

Random Walker is an image segmentation algorithm which has a walker move around an image. Several seed points with an annotated label are spread across the image at initialization, and when the walker reaches a seed point, the pixel it started from will then be annotated with the corresponding label. The transition probabilities between pixels are based on the pixels' similarity, that is, if two pixels have the same intensity a transition to that pixel is likely. In an iterative algorithm, several walkers are simulated and the pixel will be annotated the class it reached the most times. Other ways to simulate the algorithm exists, an example would

be to solve an anisotropic diffusion equation.
The Random Walker approach uses *skimage.segmentatio.random_walker* to make the noisy output from the segmentation more structured. As the skimage function expects an image the predicted segmentation is stacked on top of itself. This makes a 'up or down' transition for the Random Walker likely, but we are only interested in whether the Random Walker reaches the left or right 'end' where the seed points for a healthy and inflamed classification resides, and because the 'left/right' transition probabilities remain the same, this approach will yield the correct result. After applying the Random Walker approach we expect the new video segmentation to contain larger and more continuous blocks and ideally only two blocks giving us the predicted separation point.

4.3.2 Scoring heuristic

The scoring heuristic scores each 'transition' between the segmentations of a video based on the confidence of the predictions. For each frame, the confidence of that prediction (the output probability for the most likely class) is multiplied by the confidence of the frame immediately to the right. This gives a score between 0.25 and 1.0 to each 'transition' between the frames. This gives a high score if the model is confident in its prediction of two frames next to each other, but a low score if it is insecure. When each transition have been scored, the separation point is chosen as the point with the lowest score. If several transitions share a lowest score, the median is taken to choose a separation point which actually have a lowest score, rather than likely choosing a random point by using the mean.

4.4 Treatment prediction

When doing treatment predictions we have a total of five classes as four different treatments were prescribed to the patients in the datasets, along with some patients being healthy. The treatment prescribed is highly correlated with how much of the colon and which parts of the colon is experiencing inflammation. When doing segmentations of the videos, this is essentially the information we get, and it is thus attempted to predict the treatment from these segmentations. This means, based on a video segmentation we will predict which of the five classes of treatment the patient received.
To obtain the desired video segmentations, the best performing 2D ResNet model is used on each of the 30 suitable videos for this task. This would ideally give us one continuous block of inflammation predictions followed by one continuous block of healthy predictions for each video, which would be similar to what a doctor would observe. Because the videos vary in lengths, a vector consisting of 1000 evenly spaced samples is made from each of the 30 segmentations. These 30 vectors of size 1000 are then used to predict on.
For this task a 1D ResNet is used to process the vectors. Modifications are made similar to the 2D ResNet previously used, and can be seen in Figure 8. 'One-vs-rest' binary classifiers were also trained to make ROC curves. For these models the last fully connected layer reducing the feature size to 5 was replaced with a fully connected layer reducing the feature size to 1 and the softmax activation function was replaced with a sigmoid activation function.

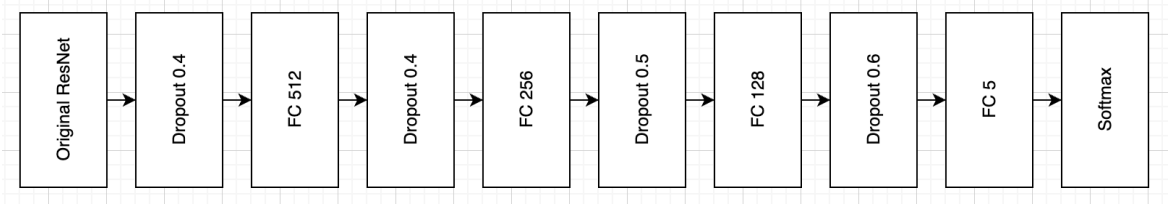


Figure 8: Modified 1D ResNet model used for treatment predictions. The modifications lies in the added fully connected layers after the original fully connected layer of size 1000, and the added dropout layers.

4.5 U-Net for video segmentation

As another approach to get video segmentations for the treatment classification, a 1D U-Net was used to predict the video segmentations. This approach was used to reduce the noise of the 2D ResNet predictions and thus to provide more accurate video segmentation vectors for treatment predictions.
The segmentation results of the 2D ResNet was used as input for the 1D U-Net model and the true segmentation was used as target vectors. Given the videos vary in length a vector consisting of 1000 evenly spaced samples is made from the results of the 2D ResNet and the target vector.

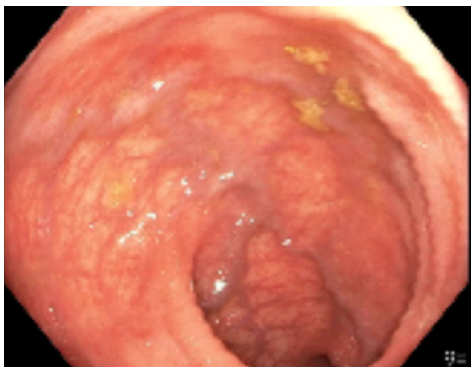
4.6 U-Net predictions for treatment prediction

With the refined U-Net segmentations, treatment predictions were then again attempted using the same approach as previously presented, but now using the U-Net segmentations as input. The same 1D ResNet models from subsection 4.4 were used again.

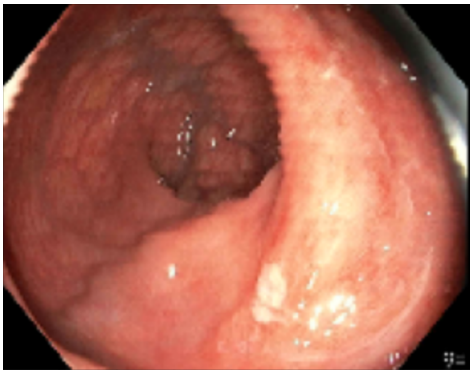
5 Results

In this section the results of the thesis work is presented. First we will take a look at some examples images from two endoscopy videos to familiarize ourselves a little with the data the models have been trained on. We will then look at the model selection and how the best performing model was chosen, followed by the separation point predictions of the post-processing approaches. Next we look at the treatment predictions from the segmentation results of the 2D ResNet, before we try and refine these results by a using U-Net model. To end with, we use the U-Net segmentations for treatment predictions.

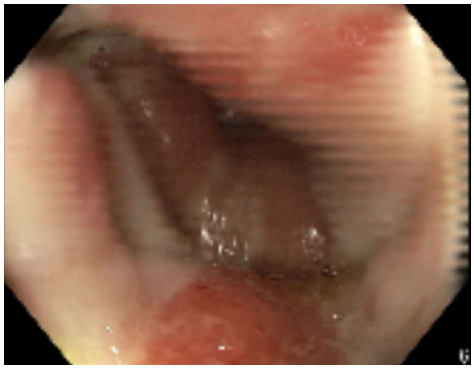
In Figure 9 we see four examples of training data from two different endoscopy videos and two different patients. In the left column we have examples of inflamed bowel tissue and in the right column examples of healthy bowel tissue. The top row of images is from the same video and likewise, the bottom row of images is from the same video. We note how even though the image pairs comes from the same patient, the colouring of the bowel tissue can change quite a lot throughout a video, which especially is seen in the bottom row of images. We also note, how much the colon twists has a large impact on the lighting, and thus on the colouring of the tissue. And lastly we also see from image (a) and (d) how similar healthy and inflamed tissue can look.



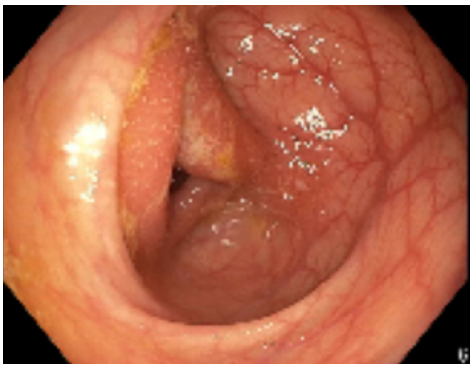
(a) Inflamed example image from validation video.



(b) Healthy example image from validation video.



(c) Inflamed example image.



(d) Healthy example image.

Figure 9: Examples images from two endoscopy videos.

5.1 Model evaluation and selection

Training was conducted by training several 2D ResNet models, one for each of the datasets, and using a single video as a constant validation set to tune learning rate, weight decay and the dropout layers. For loss functions binary cross entropy was used and for optimization Adam was used. Each model was trained for 65 epochs with a learning rate of 10^{-5}. No weight decay was added. The true segmentations was 'artificially' constructed by concatenating a vector of zeros equal in length to the number of frames up to the true separation point, and a vector of ones equal in length to the remaining frames of the video, such that we get one continuous

block of inflammation predictions followed by one continuous block of healthy predictions which correspond to how the doctor annotated the video.
In this section two results will be reported. First a five fold approach used to evaluate the model performance where each dataset is split in five folds, then a model is trained on four folds and evaluated on the constant validation set. Second, models are trained on all the data from each dataset respectively, and then used to predict each frame in the validation set for visual inspection.

Dataset	Fold 1		Fold 2		Fold 3		Fold 4		Fold 5		Avg. T	Avg. V	Frames
	T	V	T	V	T	V	T	V	T	V			
Idx_4_skip_10	100.0	67.9	100.0	33.2	100.0	35.5	100.0	43.1	100.0	45.9	*100*	*45.12*	499
Idx_4_5_skip_20	99.2	47.2	100.0	41.0	99.8	35.6	100.0	39.0	99.5	33.0	*99.7*	*39.16*	500
Idx_2_3_4_5_6_skip_50	99.5	71.4	99.5	63.7	99.5	70.0	99.7	45.5	100.0	43.4	*99.64*	*58.8*	491
Idx_2_3_4_5_6_skip_20	100.0	60.9	99.7	61.8	99.7	66.2	99.7	71.4	99.8	65.5	*99.78*	*65.16*	1226
Idx_4_14_18_20_32_skip_20	99.7	33.9	100.0	42.6	99.9	40.6	99.9	37.2	99.8	38.2	*99.86*	*38.5*	997
Idx_4_14_18_20_32_skip_5	99.9	69.3	100.0	57.1	86.4	39.5	99.5	55.9	99.9	60.9	*97.14*	*56.54*	3978
Idx_3_23_skip_10	100.0	72.0	100.0	72.3	100.0	72.6	100.0	72.4	100.0	70.6	*100.0*	*72.0*	829
Idx_19_24_skip_5	100.0	26.8	100.0	25.9	99.8	27.7	100.0	27.3	100.0	26.7	*100.0*	*26.9*	600

Table 1: Evaluation results for training a modified 2D ResNet on different datasets. All results are in percent. At each fold, T stands for training accuracy and V stands for validation accuracy.

In Table 1 we see the evaluation of the five fold approach. T stands for training accuracy and V stands for validation accuracy. All results are reported in percent. We note dataset *Idx_3_23_skip_10* and *Idx_2_3_4_5_6_skip_20* have the highest validation accuracies while *Idx_19_24_skip_5* has the lowest followed by *Idx_4_14_18_20_32_skip_5*. We also note there seemingly is no relation between the sheer number of training images and high validation accuracy, but there is some relation when we chose to add more images from the same dataset. Likewise there is seemingly no relation between the different number of patients the training images comes from and high validation accuracy.

To assert whether the reported accuracies are accurate, we will now take a look at how the models predicted on some videos. First we will look at how each of the models predicted on the validation video to establish a baseline. The results can be seen in Figure 10 and in Figure 11. For each illustration the true segmentation of the video has been drawn as the first bar. For these results, this means approximately 72% of the frames are annotated as inflamed and the last frames as healthy. The second bar is the model predictions. The blue line indicates the models' confidence in a healthy prediction, and a probability above 50% means it has predicted the frame is healthy.

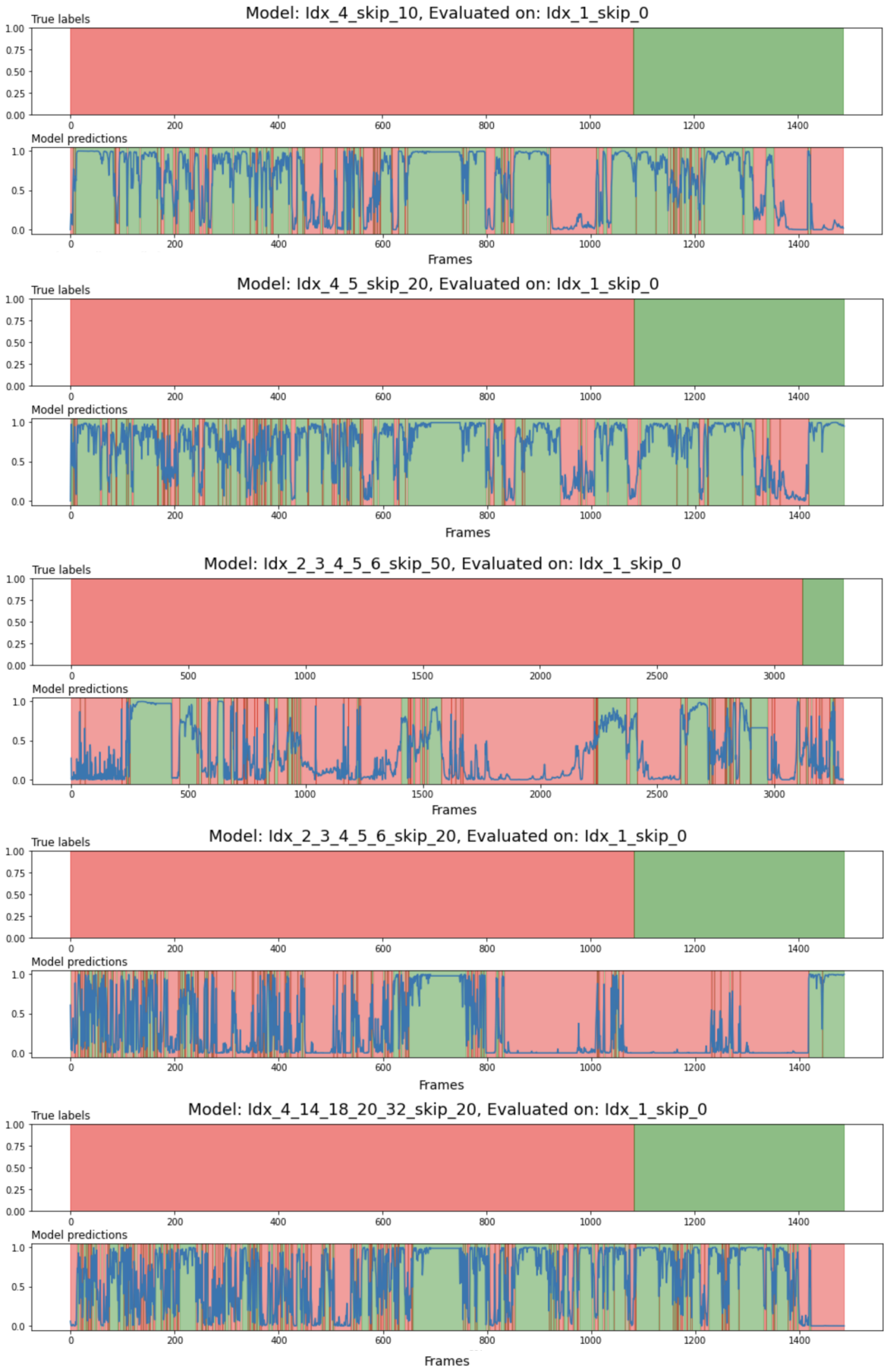


Figure 10: How the first five models predicted on the validation video.

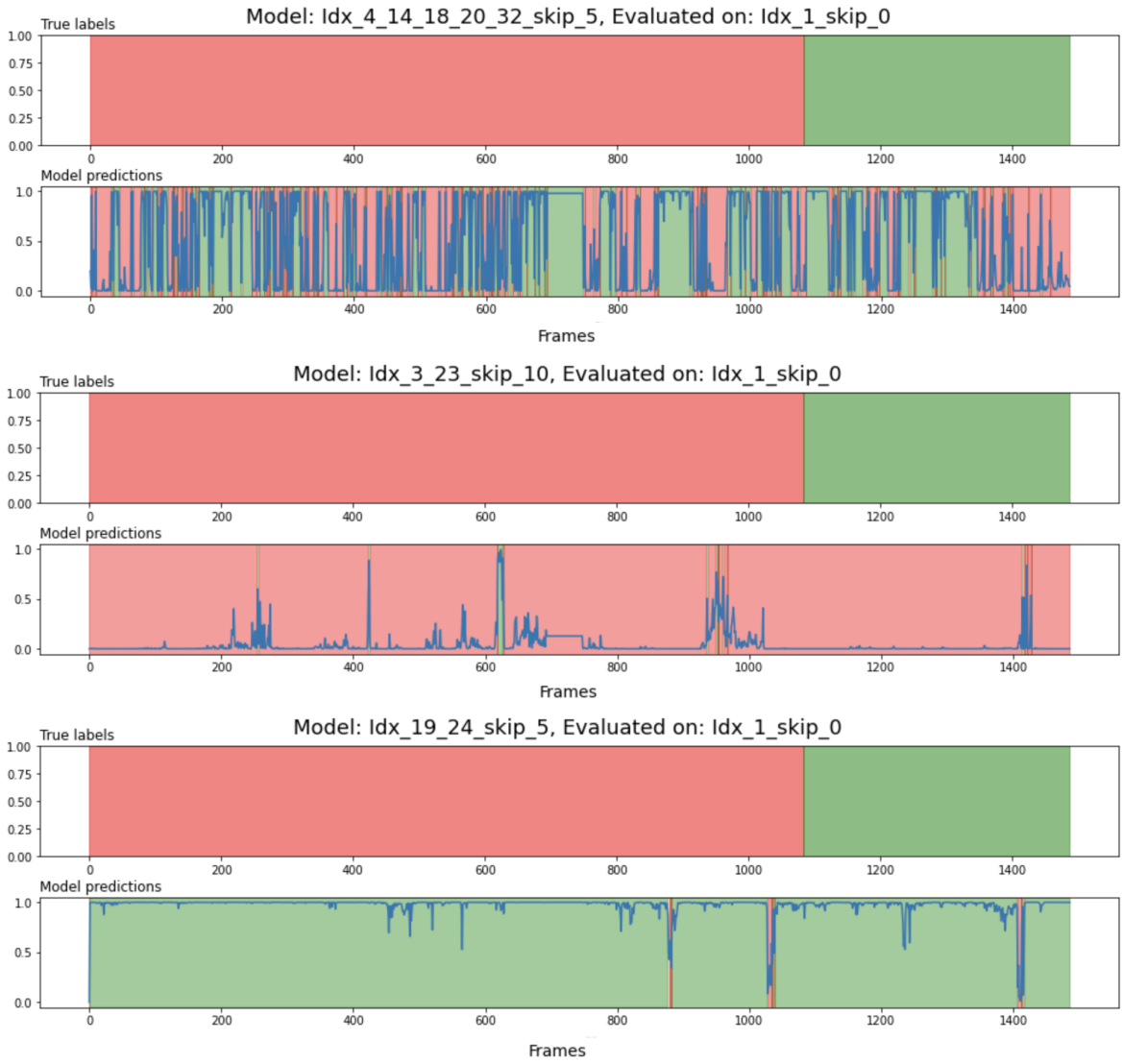


Figure 11: How the last three models predicted on the validation video.

We note how models *Idx_4_skip_10* and *Idx_4_5_skip_20* predict overly many healthy frames but seem to be fairly confident in their predictions. Model *Idx_2_3_4_5_6_skip_50* have comparably predicted a lot more inflamed frames, but it still seem to be confident some of the first frames are healthy. Looking at model *Idx_2_3_4_5_6_skip_20* which scored the second highest validation accuracy, we see it has some rather volatile predictions on the early frames where most however seem to be classified as inflamed. On the other hand, it seem to predict quite confidently too many inflamed frames towards the end of the video. Model *Idx_4_14_18_20_32_skip_20* is the first model trained across several patients, but this models seem to predict a mix of the previous models as it has some volatile predictions in the beginning of the video, but towards the middle it begins to confidently predict overly many healthy frames. Model *Idx_4_14_18_20_32_skip_5* is the second model to be trained on several patients and also had a significant larger training set than the other models. Here we see only a few blocks of consecutive healthy predictions and otherwise some very volatile predictions. Towards the end of the video it predicts a large part correctly healthy, but it also predicts the very end inflamed. Model *Idx_3_23_skip_10* was the model which achieved the highest validation accuracies, but we also see it very rarely predicts other than inflamed. As the validation video

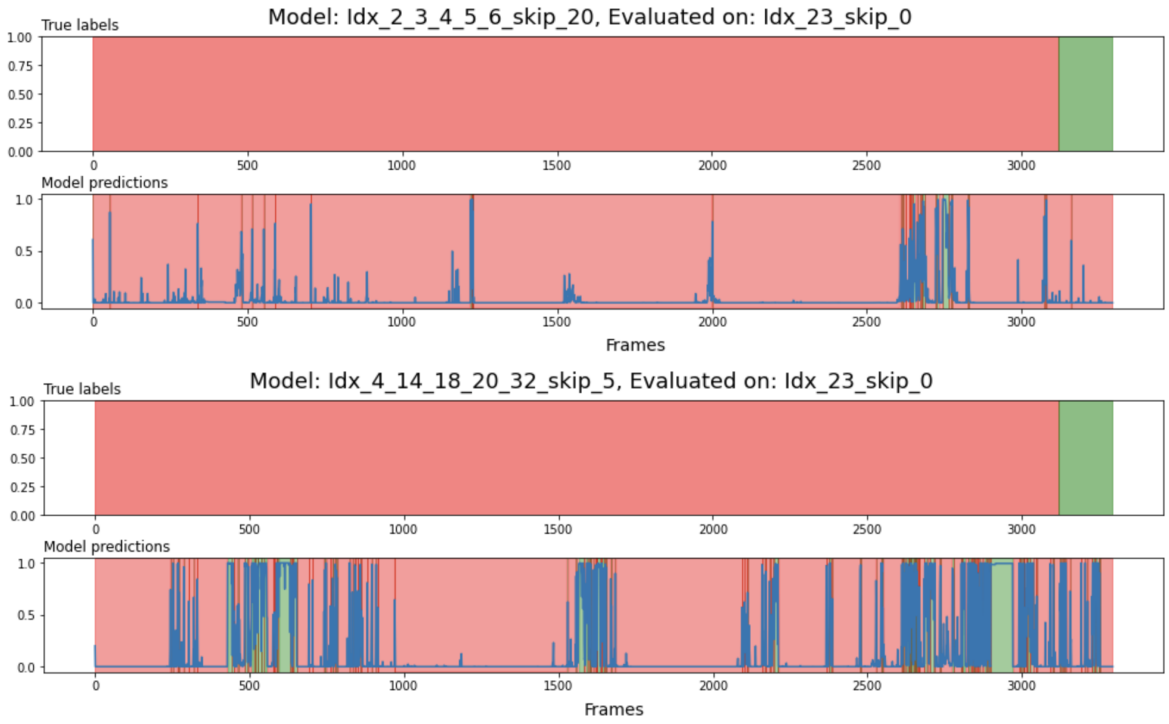


Figure 12: How the two selected models perform on test set *Idx_23_skip_0*.

consists of approximately 72% inflamed frames, it would seem its accuracy is skewed because of the imbalance in the data. Lastly we have model *Idx_19_24_skip_5* which has the opposite issue where it only predicts healthy.

To select which model to move forward and perform treatment predictions with, models *Idx_2_3_4_5_6_skip_20* and *Idx_4_14_18_20_32_skip_5* were selected to predict on more videos. *Idx_2_3_4_5_6_skip_20* was selected as it seem to predict the best on the validation video and *Idx_4_14_18_20_32_skip_5* because prior knowledge say models trained on more and varied data should perform better, and thus we verify its validation results from Table 1 are not amiss.

In Figure 12 we see the predictions of the two models on an unseen video which is biased towards being inflamed. We see model *Idx_2_3_4_5_6_skip_20* predicting confidently and almost exclusively inflammation with a few healthy predictions in the wrong places. Model *Idx_4_14_18_20_32_skip_5* is more uncertain, and predicts healthy in some spots while having oscillating predictions. The largest consecutive part where it confidently predicts healthy is when it should have predicted inflammation.

In Figure 13 we see the predictions of the two selected models on an unseen video biased towards being healthy. We see model *Idx_2_3_4_5_6_skip_20* still confidently predicting most frames inflamed, with a few correct exceptions towards the middle of the video. Model *Idx_4_14_18_20_32_skip_5* predicts most of the video inflamed too, but is not as certain and also correctly predicts one large block healthy.

As a last validation, in Figure 14 we see the predictions of the two selected models on an unseen video biased towards inflammation. Model *Idx_2_3_4_5_6_skip_20* is rather volatile in its predictions at the beginning of the video, but seem to correctly predict mostly inflammation. Towards the end it seem to capture some of the healthy frames, but predicts a few too many. Model *Idx_4_14_18_20_32_skip_5* seems on the other hand very volatile, and seem to have some passages where it wrongly predicts healthy. Towards the end we still see some volatile predictions, but it does not seem to capture the healthy frames at the end.

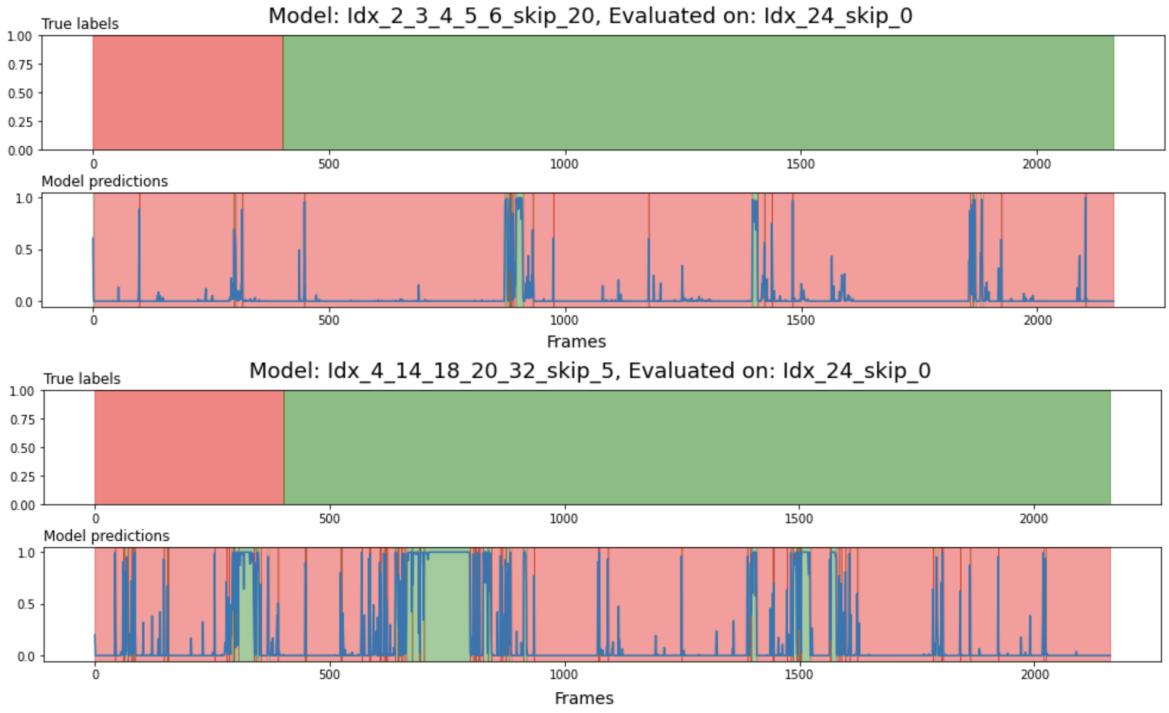


Figure 13: How the two selected models perform on test set *Idx_24_skip_0*.

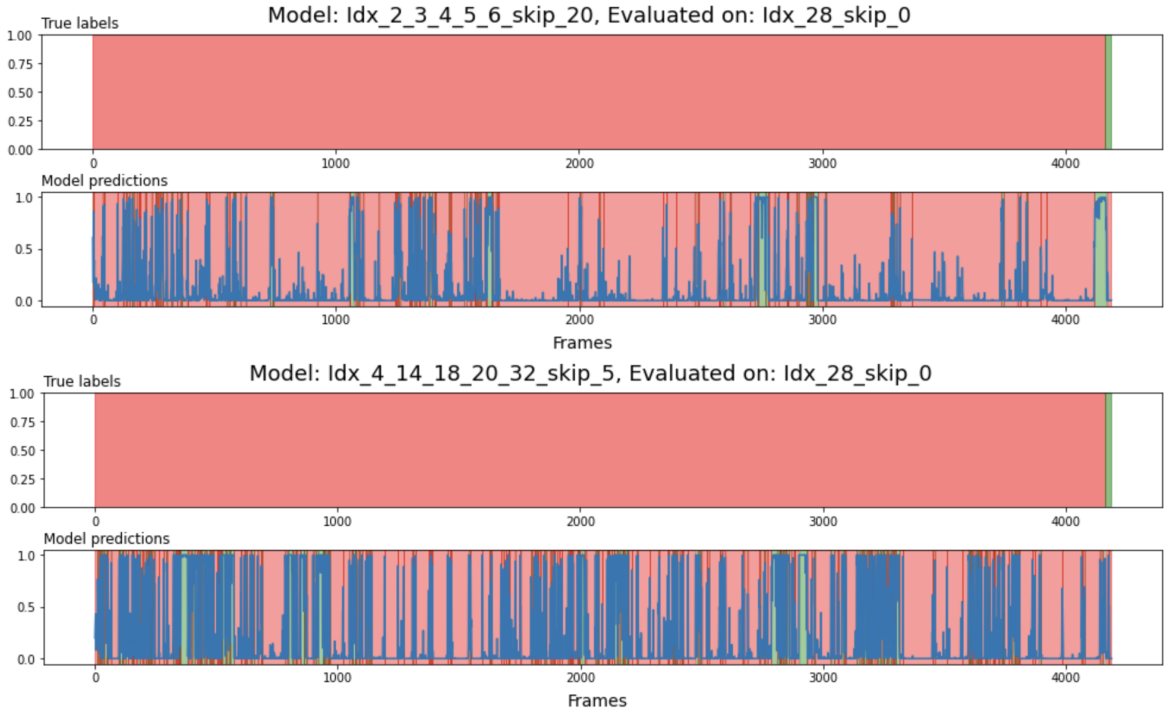


Figure 14: How the two selected models perform on test set *Idx_28_skip_0*.

Based on these results model *Idx_2_3_4_5_6_skip_20* is considered to be the most accurate, and will be used moving forward towards predicting separation points and treatments.

5.2 Separation point prediction

Having decided on a model to move forward with, we can now evaluate how well the post-processing approaches performed by looking at their separation point predictions.

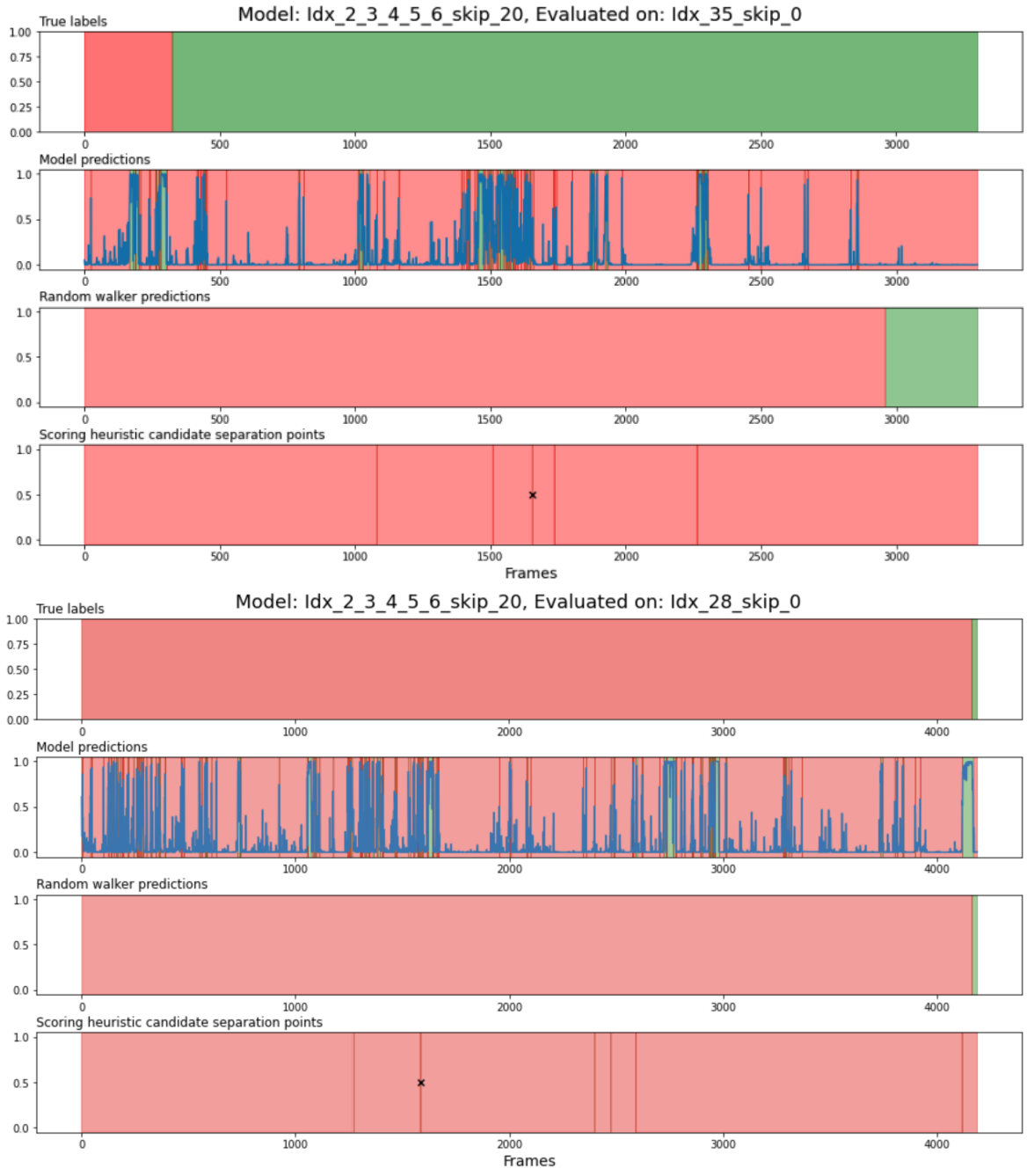


Figure 15: Two separation point results.

In Figure 15 we see two evaluations, each consisting of four bars where the top bar is the true segmentation, the second is model prediction, third is the result of applying the Random Walker post-processing and the fourth is the result of applying the scoring heuristic. It should be noted the scoring heuristic predicts candidate points and one 'believed separation point' marked with an 'x'. This means, the marked points are 'transitions' which has a score less than 0.3, and the 'x' marks the median of the lowest scoring 'transitions'. The scoring heuristic bar should thus not be read as a segmentation, but as several possible and one believed separation point.

Given the two examples in Figure 15 we see a tendency for the Random Walker to predict a 'high' separation point, whereas the scoring heuristic is rather insecure due to the insecure model predictions. These are general trends which also can be seen in Table 2 where the results for all suitable videos has been reported. Videos 2 through 6 has not been reported as the model was trained on these.
In Table 2 we see the true separation points (row 'True'), the predicted separation points for both post-processing approaches (row 'SH P' and 'RW P'), how far each prediction was from the true separation point measured in absolute number of frames (row 'SH #' and 'RW #'), and how far each prediction was measured in percentage of the total number of frames in the video (row 'SH %' and 'RW %').

idx	1	7	8	9	10	11	12	13	14	15	16	17	18	19	20
True	1086	3303	241	0	0	0	0	0	1575	3300	3299	3301	4164	76	1026
RW P	1459	3279	202	4979	4644	3300	3281	3291	3291	3272	3260	3255	4167	803	2739
SH P	43	2345	107	614	2076	2895	1354	3179	3179	887	572	1400	4153	374	444
RW #	373	24	39	4979	4644	3300	3281	3291	1716	28	39	46	3	727	1713
SH #	1043	958	134	614	2076	2895	1354	3179	1604	2413	2727	1901	11	298	582
RW %	0.25	0.01	0.16	1.00	0.93	0.66	0.99	1.00	0.52	0.01	0.01	0.01	0.00	0.87	0.52
SH %	0.70	0.29	0.56	0.12	0.42	0.58	0.41	0.96	0.49	0.73	0.83	0.58	0.00	0.36	0.18

idx	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
True	3295	3299	3119	402	0	0	0	4164	0	0	0	0	3299	3302	325
RW P	2446	3261	3270	1880	4978	4779	4674	4167	49	3259	3293	3261	3296	3295	2958
SH P	2441	543	1221	1428	1672	2113	4672	4153	53	2202	2586	3052	2562	1614	1653
RW #	849	38	151	1478	4978	4779	4674	3	49	3259	3293	3261	3	7	2633
SH #	854	2756	1898	1026	1672	2113	4672	11	53	2202	2586	3052	737	1688	1328
RW %	0.26	0.01	0.05	0.68	1.00	0.96	1.00	0.00	0.88	0.99	1.00	0.99	0.00	0.00	0.80
SH %	0.26	0.84	0.58	0.47	0.34	0.42	1.00	0.00	0.95	0.67	0.78	0.93	0.22	0.51	0.40

Table 2: Separation point predictions for all suitable videos. Each video has been enumerated and given and index to be identified with. The 'True' row depicts the true separation point. 'RW P' is the separation point predicted by the Random Walker. 'SH P' is the predicted separation point for the scoring heuristic (the believed point). 'RW #' is the absolute number of frames to the true separation point from the Random Walker prediction. 'SH #' is the absolute number of frames to the true separation point from the scoring heuristic prediction. 'RW %' is the number of frames from the Random Walker prediction to the true separation point measured as a percentage of the total number of frames in the video. 'SH %' is the number of frames from the Scoring heuristic prediction to the true separation point measured as a percentage of the total number of frames in the video.

Taking a closer look at the results in Table 2, we see the average distance in frames from predictions to true separation points for the Random Walker approach is 1788.6 frames, while for the scoring heuristic this is 1614.57 frames. However, if we compute the distances in percentages of the total number of frames Random Walker comes out with an average of 51.85% and the scoring heuristic with 51.87%.
Taking a look at the distance between the predictions and true separation points in terms of frames for the Random Walker approach (row 'RW #') we see a quite large variance, as some of the videos with a lot of inflamed frames are predicted very accurately, while most of the videos with a lot of healthy frames are predicted very poorly. In general we see the Random Walker approach being very conservative, and predicting 'high' separation points.
Looking at the distance between the predictions and true separation points for the scoring heuristic (row 'SH #'), we see a lower variance as the scoring heuristic is more willing to predict lower separation points on the videos with many healthy frames, however, it is not as accurate on the videos with a lot of inflamed frames as

the Random Walker approach is.

5.3 Treatment prediction

Because only 30 data vectors were available and model *Idx_2_3_4_5_6_skip_20* was trained on five datasets, only 25 data vectors were available for predicting treatments. Two slightly different approaches of training was thus attempted. First, training was conducted by a 'leave-one-out' strategy where a 1D ResNet was trained on 24 data vectors and tested on the last one, while rotating which data vector was left out, and training a new model for each rotation. In the second approach, the true segmentations was included in training which would give 48 training vectors. The same leave-one-out strategy was deployed, but only the predicted segmentation would be used for testing. Each model was trained for 120 epochs, using cross entropy as loss function and Adam for optimization with a learning rate of 10^{-4} and with a weight decay of 0.07.

Idx	1	7	8	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
Predictions	2	0	0	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	0	0	2	2	0	2	2
True	1	3	3	4	2	2	2	1	1	2	2	2	2	1	0	0	0	1	0	0	0	0	2	2	2

Table 3: Training a model only on the predicted segmentations, we find the following treatment predictions and true treatments shown for each video. The treatment classes are: 0 being 'healthy', 1 being 'local 5-ASA', 2 being 'Oral steroid', 3 being 'IV steroid' and 4 being 'oral 5-ASA'.

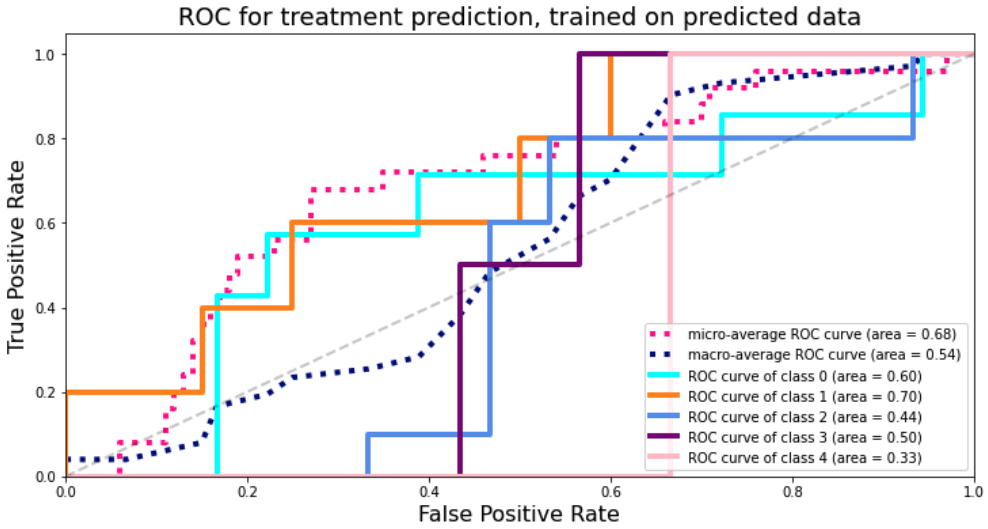


Figure 16: ROC curves and area under the curve for the five binary 'one-vs-rest' classifiers along with micro- and macro-averages. Training was performed only on predicted segmentations.

In Table 3 we see the predicted treatments along the true prescribed treatment, when we train only on predicted segmentations from model *Idx_2_3_4_5_6_skip_20*. Computing the accuracy of this approach we get 40%. However, we note the model only have predicted class 2 and class 0, and there is a heavy class imbalance in the data, with only two examples of class 3 and one example of class 4.
It can be hard to assert the quality of a multi class classifier based on accuracy, and thus five binary 'one-vs-rest' classifiers was trained with the same parameters as the original multi class classifier. These binary models was then used to draw ROC curves for which the area under the curves could be computed. The results can be seen in Figure 16.

In Table 4 we see the treatment predictions and true prescribed treatment for each video for a model trained on the predicted segmentations from model *Idx_2_3_4_5_6_skip_20* and the true segmentations. This model

was trained with the same parameters as the previous treatment prediction model. Computing the accuracy we get 48%, but we again note only class 0 and class 2 is predicted.

Idx	1	7	8	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
Predictions	2	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	2	0	0	2	2
True	1	3	3	4	2	2	2	1	1	2	2	2	2	1	0	0	0	1	0	0	0	0	2	2	2

Table 4: Training a model on both the predicted segmentations and true segmentations, we find the following treatment predictions and true treatments shown for each video. The treatment classes are: 0 being 'healthy', 1 being 'local 5-ASA', 2 being 'Oral steroid', 3 being 'IV steroid' and 4 being 'oral 5-ASA'.

In Figure 17 ROC curves are drawn for each of the five binary 'one-vs-rest' classifiers, trained with the same parameters as the original multi class classifier and trained on both predicted and true segmentations.

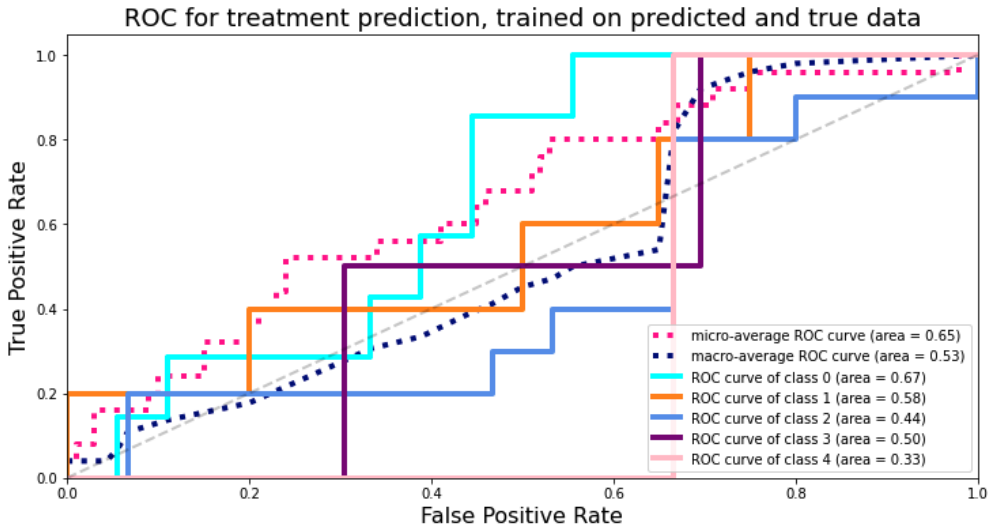


Figure 17: ROC curves and area under the curve for the five binary 'one-vs-rest' classifiers along with micro- and macro-averages. Training was performed on both predicted and true segmentations.

5.4 U-Net for video segmentation

In an attempt to make the predicted video segmentations from the 2D ResNet model more structured, a 1D U-Net was trained on the predictions of model *Idx_2_3_4_5_6_skip_20*. Training was conducted in a 'leave-one-out' fashion on the 25 predicted segmentations suitable for treatment predictions. The model was trained for 100 epochs with a learning rate of $3 \cdot 10^{-6}$, binary cross entropy as loss function, Adam for optimization and 0.25 weight decay.

The true segmentations was used as target, and was 'artificially' constructed by concatenating a vector of zeros equal in length to the number of frames up to the true separation point, and a vector of ones equal in length to the remaining frames of the video, such that we get one continuous block of inflammation predictions followed by one continuous block of healthy predictions which correspond to how the doctor annotated the video.

In the following, four predictions will be reported to show the effect of the trained U-Net model. In each illustration the true segmentation will be shown as the top bar, the 2D ResNet prediction will be shown as the second bar, and the 1D U-Net prediction and its confidence in a healthy prediction will be shown as the third bar.

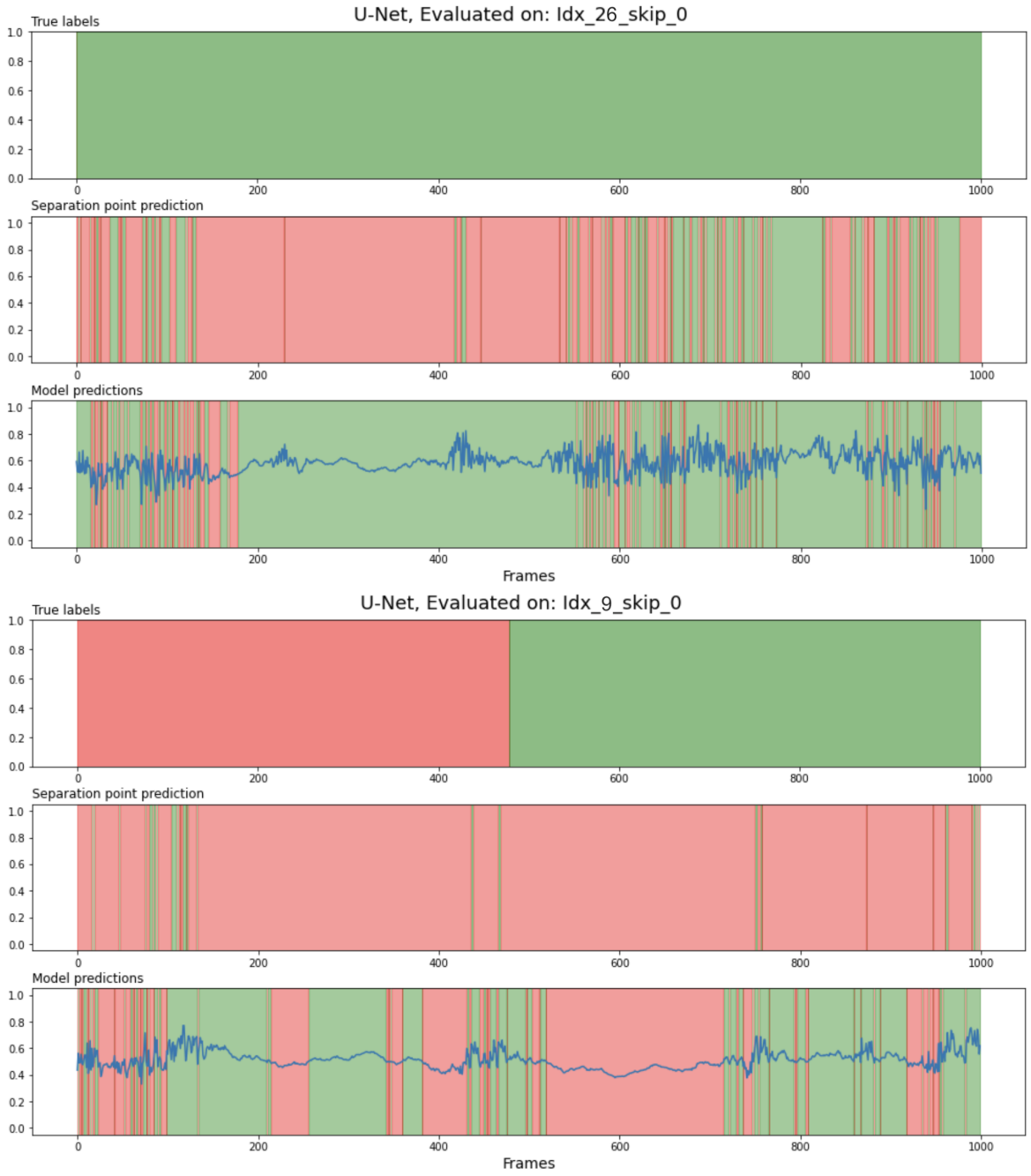


Figure 18: Results of 1D U-Net model.

Taking a look at the first result presented in Figure 18, we see the true segmentation is all healthy, but the ResNet prediction has a lot of inflammation predictions. Although the U-Net model is not confident in its predictions, it is certain in the sense it does not have very big oscillations. We also see the U-Net predictions have overturned a lot of the inflammation predictions, and made the prediction a lot more representative to the true segmentation.

Looking at the second prediction in Figure 18, we see the true segmentation being almost half inflammation and half healthy. The ResNet predictions are however almost exclusively inflammation. Again we see the U-Net predictions being very steady and overturning a lot of the inflammation predictions, however this time also making correctly predicted inflamed frames healthy.

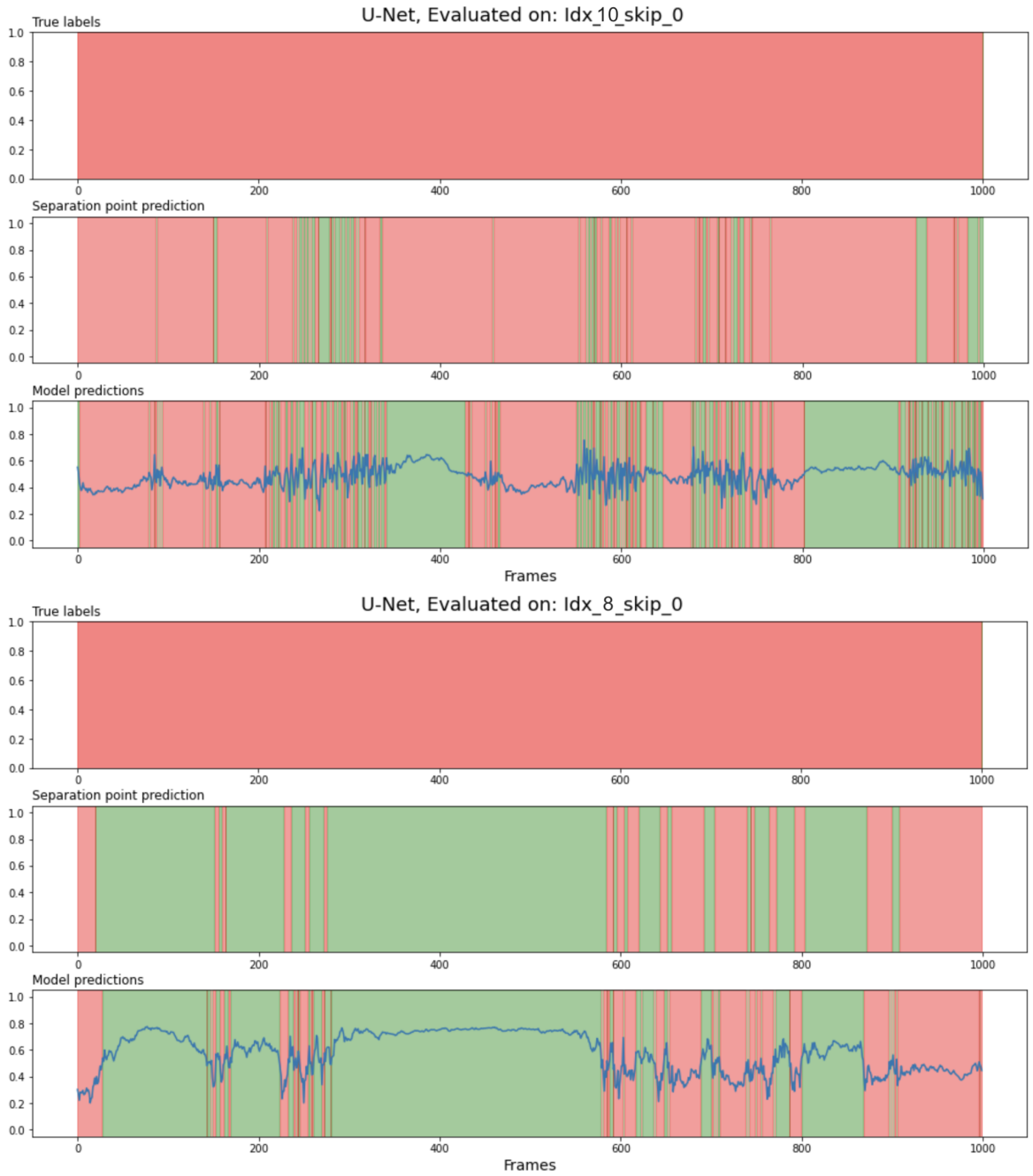


Figure 19: Results of 1D U-Net model.

This tendency of making inflamed frames healthy seem to continue as we see in Figure 19. For the first prediction we see the U-Net model likes to add healthy predictions even if they are wrong. In the second prediction we see the U-Net model is quite reluctant to add inflammation predictions when the ResNet model predicts too many healthy frames.

Even though we have only looked at four predictions this is the general tendency of the U-Net predictions. The model adds healthy predictions to the ResNet predictions, and it is very reluctant to add inflammation predictions, while being very steady and not overly confident in its predictions. As the ResNet was overly predicting inflammation, the U-Net seems to have caught on to a correct tendency of missing healthy frames. Although it has not quite managed to make its predictions two continuous blocks, its predictions are not heavily oscillating, and thus seem to have have caught on to the continuity a little.

5.5 U-Net predictions for treatment prediction

With the new segmentations produced by the U-Net model, treatment prediction was attempted again. This time the same 1D ResNet model was used, but training consisted of 70 epochs with a learning rate of 10^{-4} and with 0.01 weight decay. Cross entropy was used as loss function and Adam was used for optimization.

Idx	1	7	8	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
Predictions	0	2	1	2	2	2	1	1	1	1	1	2	1	1	1	0	1	1	1	0	2	0	2	2	2
True	1	3	3	4	2	2	2	1	1	2	2	2	2	1	0	0	0	1	0	0	0	0	2	2	2

Table 5: Training a model only on the predicted U-Net segmentations, we find the following treatment predictions and true treatments shown for each video. The treatment classes are: 0 being 'healthy', 1 being 'local 5-ASA', 2 being 'Oral steroid', 3 being 'IV steroid' and 4 being 'oral 5-ASA'.

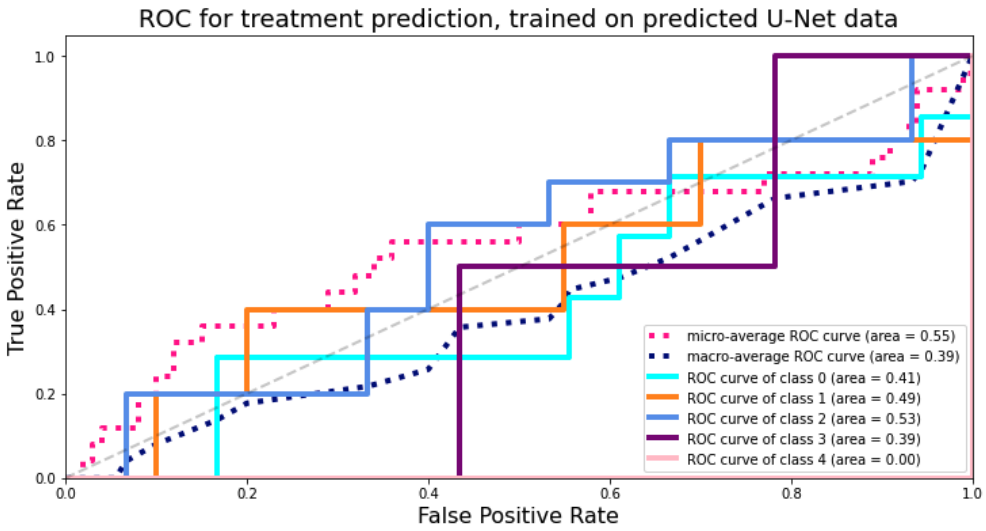


Figure 20: ROC curves and area under the curve for the five binary 'one-vs-rest' classifiers along with micro- and macro-averages. Training was performed only on predicted segmentations.

In Table 5 we see the predicted treatments along the true prescribed treatment when we train only on the predicted segmentations from the U-Net model. In contrast to the previous results we now see predictions of classes $0, 1$ and 2, where we previously did not see any predictions of treatment 1. We also see an improvement in accuracy as we now have 52%.

Five binary 'one-vs-rest' classifiers were also trained for these predictions, and they were again trained using the same parameters as the multi class classifier. In Figure 20 we see ROC curves for these binary classifiers, and we note the area under the curves being either 0.5 or 0.4 for all classes except class 4 which has 0.0, which is significantly worse than for the previous treatment predictions.

When we include the true segmentations to the training data and repeat the experiment with the same parameters, we find the results seen in Table 6. Here we again see the model predicting three classes, however, the accuracy drastically drops to 20%.

Idx	1	7	8	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
Predictions	2	2	2	2	1	1	1	2	2	2	1	1	2	2	1	2	2	2	2	2	1	0	2	1	2
True	1	3	3	4	2	2	2	1	1	2	2	2	2	1	0	0	0	1	0	0	0	0	2	2	2

Table 6: Training a model on the predicted U-Net segmentations and the true segmentations, we find the following treatment predictions and true treatments shown for each video. The treatment classes are: 0 being 'healthy', 1 being 'local 5-ASA', 2 being 'Oral steroid', 3 being 'IV steroid' and 4 being 'oral 5-ASA'.

In Figure 21 we see the ROC curves when we add the true segmentations. Here, however, we see the area under the curve for class 0 being a fair bit above 0.5, indicating predictions for class 0 are not random, while classes 1 and 2 are slightly above 0.5.

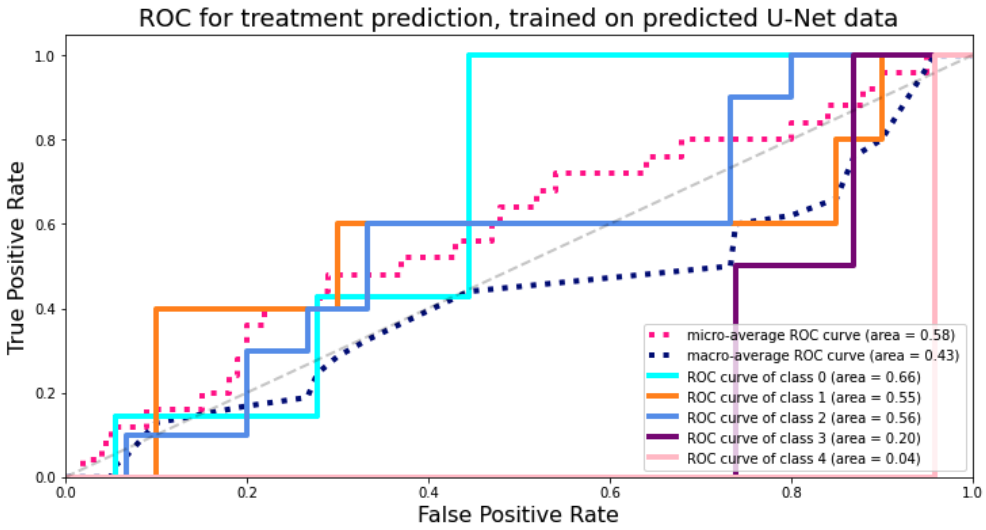


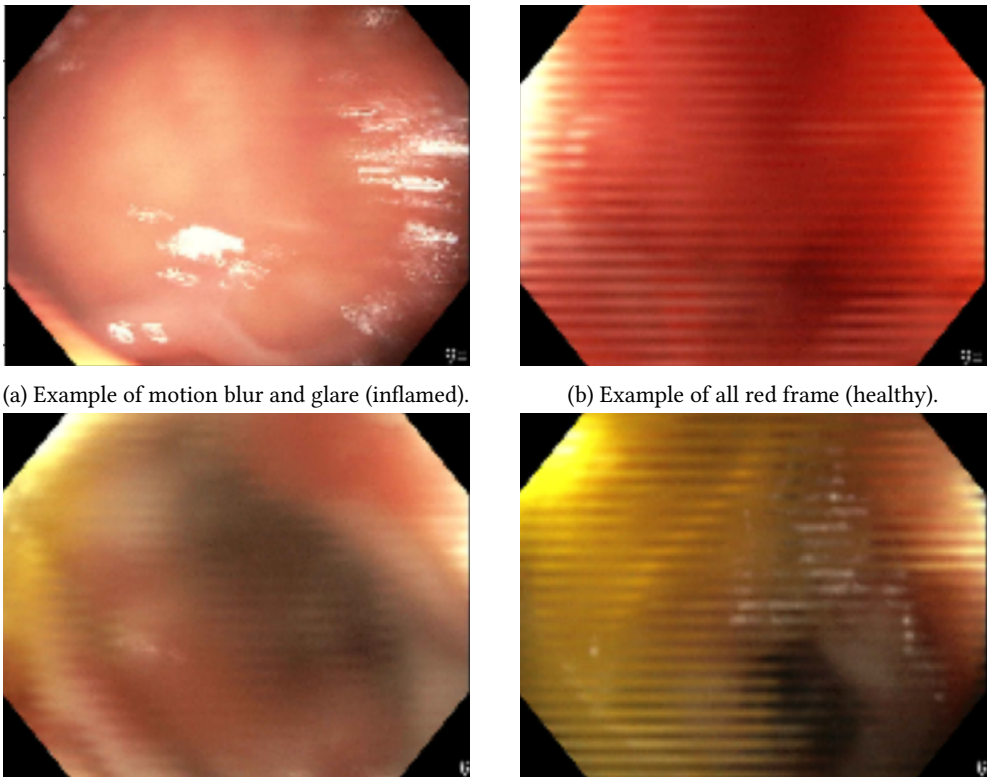
Figure 21: ROC curves and area under the curve for the five binary 'one-vs-rest' classifiers along with micro- and macro-averages. Training was performed only on predicted segmentations.

6 Discussion of results

6.1 Model evaluation

In Table 1 we see the model evaluation of models trained on the different datasets. We see the training accuracies for all models being close to 100% while the validation accuracies are lagging behind. The gap between training and validation accuracy, and the training accuracy being this high, would indicate that the model is overfitting the training data, which is likely also the reason a lot of models overly predicts either healthy or inflamed. It would thus have been sensible to add more regularization to the models for instance in the form of weight decay. This was also attempted, however, the combination of high dropout and weight decay seemed to halt learning to a degree where both training and validation accuracy would stay around 50%. Because of this, it was decided not to add weight decay, and due to time constraints towards the end of the thesis, this issue was never re-visited. As these models are the very foundation of the project, more should probably have been done to tune them more accurately. This could have been done by training for more epochs and accordingly lower the learning rate. In state of the art applications, models are often trained for several days with low learning rate, however, such a setting would not have been feasible for this project as models was run in Google Colab with very limiting usage restrictions. Models could, however, have been trained for a lot longer than they were with a much smaller learning rate. Different model architectures could have been attempted. This could both be a different variant of ResNet, but also a completely different architecture. Also more combinations of dropout and weight decay could have been attempted to resolve the no learning issue. Likewise, different combinations of skipping frames in the videos used for training could have been explored more.
Splitting all the videos into seven folds, training on six and validating on one was also attempted. This was attempted because, usually the more diverse and the larger a volume of data used for training leads to the best results when training machine learning models. However, this was extensively tested by varying epochs, learning rate, number of skipped frames, weight decay and dropout. Because the validation results would not move above 56% the experiment was prematurely ended, and thus not reported.

An interesting tendency we see in Table 1 and when it was attempted to split all the videos into seven folds is that, the sheer number of frames does not seem to have any relation to the model accuracies. Although it is not directly the sheer number of training examples which is the reason machine learning models in general perform better on larger datasets, but the increased likelihood of having varied data, we would still have expected to see higher accuracies when training on more data and across several patients. To some extent we do see more data increases accuracy, as the two models trained on the same datasets but with varying amounts of data, both do increase in accuracy as their training set increases in size. However, we also see model *idx_2_3_4_5_6_skip_50* having higher accuracy than model *idx_4_14_18_20_32_skip_5* although the latter model's training set is eight times larger. This might simply boil down to the validation set being more similar to the data model *idx_2_3_4_5_6_skip_50* is trained on, but during the visual inspection on other videos from other patients, model *idx_2_3_4_5_6_skip_20* (trained on similar data) still performed better than model *idx_4_14_18_20_32_skip_5*. This also goes to show training on more patients do not necessarily increase model accuracy either.
It is hard to conclude why more data and training on more patients seemingly have no or limited effect on the model accuracy. One reason could be the features found vary too much or exists in both classes, contradicting themselves, and instead of making the model more robust, confuses it. Another reason could be the degree of inflammation varies too much, making it harder to find common features, and maybe it would have been beneficial to construct datasets with the type of treatment prescribed in mind. A last reason could be the amount of noisy images sampled from the videos vary, and the datasets trained across several patients might coincidentally have more noisy frames in them, although this has not been investigated.
Exactly why model *idx_2_3_4_5_6_skip_20* performs the best is also hard to conclude, but it is likely correlated



(a) Example of motion blur and glare (inflamed).

(b) Example of all red frame (healthy).

(c) Example of blurred frame (inflamed).

(d) Example of a large part of a frame changing colour to yellow (inflamed).

Figure 22: Examples of noisy frames in the endoscopy videos.

to the above.

Looking into the visual results in Figure 10 and Figure 11 we see some quite volatile predictions from most models. This is likely due to the nature of the videos, where many frames all throughout the videos are all black or red, having glare effects or having a significant colour change when the camera comes too close to the colon to be able to focus. This is not an issue for a doctor to see past, however, when the model is trained on these noisy frames which belongs to both the healthy and inflamed classes, it is naturally a hard decision for the model to classify them correctly. The ResNet model also has no notion of what happens before or after the frame it processes, which gives the model no context to work out if an all red frame came before or after other inflamed frames. Another issue stems from the motion of the camera as some frames are blurred, which might make it hard to detect the features needed to determine if the frame contains signs of inflammation. In Figure 9 (a) and (d) we also saw an example of two very similar frames, being of opposite classes, and one could raise the question if (a) is inflamed or the lighting simply makes it look like it. This means although it might appear the models are indecisive or having issues predicting parts of the videos due to the volatile predictions, I argue it is to some extent impossible to avoid, and should be expected. Examples of noisy frames can be seen in Figure 22.

One could remove all of these frames from the training data, making it a lot easier for the model to find the needed features in the images to detect inflammation, however, when it is then presented with a video 'from the real world', it would likely not be able to generalize due to unseen noise.

To evaluate the performance of model *Idx_2_3_4_5_6_skip_20*, we take a look at its 65% validation accuracy and at its predictions in Figure 12, Figure 13 and Figure 14. Its accuracy is by itself not impressive, and especially not when taking into consideration about 72% of the validation video are inflamed frames. Looking

at its predictions, the model overly predicts inflammation, which leads to a conservative model which has a hard time predicting correctly on videos which are balanced or biased towards healthy frames. On the other hand, the model seems to perform decent on videos biased towards inflamed frames, where it most often finds some correct healthy frames. Although we should not expect perfect segmentation due to the noisy nature of the videos, we conclude none of the models are performing optimally, and model *Idx_2_3_4_5_6_skip_20* to a small degree produces usable results, however, they are not of a quality where they can contribute with any meaningful information to a doctor.

6.2 Separation point predictions

A general tendency for the Random Walker post-processing was to predict the separation point at the location of the last uncertainty in predictions. That is, the rightmost part of the segmentation where there would be either volatile predictions or elevated probability of predicting healthy. Because the Random Walker has a hard time crossing points or frames where the model predictions are dissimilar, this is somewhat expected, but due to the random noise in the videos previously discussed, this makes this approach conservative and not very robust. This was also verified as on average, the Random Walker approach would predict a separation point 51.85% of the total number of frames in a video away from the true separation point. For the results produced by model *Idx_2_3_4_5_6_skip_20* this approach is thus not usable to predict the separation points. Had we had more accurate predictions, the results would likely have been better, however, due to the noisy nature of the videos, and how the Random Walker would likely get 'stuck' on the noise, it is still questionable if this approach can produce meaningful results.

The scoring heuristic works by finding dissimilarities between the prediction probabilities of two consecutive frames. If the there is a high probability both frames belong to the same class, the score is high, whereas, if there is a high probability they belong to different classes, the score is low. Thus, this approach should be good at finding the transitions between frames where the model is insecure or predicts on noisy data. This is also generally what we see, and often one of the candidate points are near the true separation point, however, it is rare this candidate point holds the actual lowest score. As it on average would predict a separation point 51.87% of the total number of frames in a video away from the true separation point, its 'believed separation point' is not usable to predict the separation points.
Providing candidate points could be useful for a doctor to do a quick evaluation based on the predicted segmentation and the candidate points. This could save the doctor time by not having to look the whole video through. But, as it is hard, if not impossible, to tell if any of the candidate points actually resembles the true separation point, only having the predicted segmentation available, the fact one of the candidate points often (but not always) is accurate is not of much help in practice.

6.3 Treatment predictions

While most treatment annotations are clear, some videos have been annotated with treatment 'x or y'. These cases are indices 7 and 8 which have been annotated as class 2 or 3, index 14 annotated class 2 or 4, index 28 annotated class 0 or 1 and indices 34 and 35 annotated class 1 or 2. In all these cases, the largest class number has been chosen as the true class, however, this might add or subtract a few percentages of accuracy depending on how the true classes are chosen.

In Table 3 we saw the 1D ResNet model could achieve an accuracy of 40%, which is good considering random predictions would have yielded 20% accuracy. However, if these predictions were to be used by a doctor, as for instance a second opinion, 40% is rather low and would be too unreliable. We also see the model only predicting classes 0 and 2, which would indicate the model does not generalize too well.
Looking at Figure 16 we see class 0 and class 1 having an area under the curve above 0.5 which indicates the models have some measure of class separability whereas for classes 2 and 3 we see the scores being very close

to 0.5. Looking at class 4, we have a score of 0.33 which would indicate it is quite hard to predict correctly. When we look at class 3 and 4, we need to remember only 2 and 1 examples exists in the data, and thus the model will naturally have a hard time predicting these classes. If we look at the micro average, we see a score of 0.68, which would indicate we can expect a model to have some measure of class separability when trained on this data, which we already have seen was the case. However, looking at the macro average which takes the class imbalance into consideration, we should not expect any meaningful results. This might, however, be skewed as class 3 and 4 have this few examples.

In Table 4 we added the true segmentations during training, and the accuracy increased to 48%. However, the model still only predicts class 0 and 2, indicating it overfits the training data. Looking at Figure 17 we see the scores for class 0 and 1 almost swap, and the rest of the classes and the macro average staying very similar to what we had without adding the true segmentations. The micro average falls slightly, but still is at a level where some notion of class separability would be expected. The micro average falling, and yet the multi class classifier achieving a higher accuracy could be caused by the model predictions being closer to each other, i.e. the model is not as confident in its choice of class, but still ends up choosing the correct class.

Overall it would seem more tuning could have been done to achieve better results. Although nearing 50% accuracy on a 5 class classification problem seems decent, the ROC curves are not very confident, and even the highest scores being rather mediocre. Again training for more epochs with an accordingly lower learning rate could be a possible improvement. The indications of overfitting in both results would also point towards more regularization would be needed. Adding dropout to force the models to learn a wider array of features in the data could be one choice. As we are seeing the model is capable of solving the problem to some extent, a change of model architecture is most likely not needed to raise the performance.

6.4 U-Net for video segmentation

Opposite to ResNet, that would make its predictions on each individual image without considering the images before and after it, the U-Net models were trained on the segmentations, and because U-Net works on several levels of abstractions it would be expected to capture the 'structure' of the segmentations, i.e. how the true segmentations are blocks of first inflamed frames and then healthy frames. This structure is to a large degree missing from the initial 2D ResNet predictions, and thus the hope with the results of using the U-Net, was to bring this structure into the predictions.
Looking at Figure 18 and Figure 19 we see the U-Net overly predict healthy frames, which probably is a reaction to the ResNet overly predicting inflammation. The structure we were seeking, is also only brought back to a limited extent. Because the U-Net predictions overrule the correctly predicted frames from the ResNet, it seems the problem with the original predictions have just shifted from being overly many inflamed predictions to overly many healthy predictions.

6.5 U-Net predictions for treatment prediction

Although the predictions of the U-Net models did not seem to make the results much more usable than the original 2D ResNet results, we see a big difference in how the treatments are predicted. In Table 5 we now see predictions of class 0, 1 and 2. Because of the low number of examples of class 3 and 4 treatments, it is not surprising we do not see these classes predicted. We now have an accuracy of 52% which, taking into consideration we have 5 classes, is decent. However, it is still not high enough to, for instance, act as a second opinion for a doctor. Looking at Figure 20 we now see a drastic drop in the area under the ROC curves. We now only have class 4 not being near a score of 0.5. Given how the 'one-vs-rest' classifiers struggle, the high accuracy could be caused by the multi class classifier not being very confident and having almost equal probabilities for each class, but the correct class coming out on top for the most part.

In Table 6 the true segmentations were added to the training. We still see the model predicting class 0, 1 and 2,

however, we now see the accuracy drop to 20%, indicating random guesses were made. Looking at Figure 21 we however see class 0 having a score of 0.66, suggesting these predictions were not entirely random. The rest of the classes do struggle however. It is likely adding the true segmentations simply confused the model during training, as the features found in the predicted segmentations did not match the features found for the true segmentations. Also, the drop in accuracy could stem from the multi class classifier not being confident in its predictions, giving almost equal probabilities to each class, but this time the correct classes would not come out on top.

7 Conclusion

In conclusion we saw during the model selection, that a more diverse and larger training set was not necessarily the key factors to achieve good results as we would have expected. Instead it was a model trained on videos from the same patient which was deemed the best performing model. It is hard to conclude why this is, but it is likely due to a combination of the training data being diverse, but not overly diverse, while the training data not containing overly many noisy frames. However, this model was, together with the others, heavily overfitting and more regularization should have been used during training. Nonetheless, this model had 65% validation accuracy, but overly predicting inflammation when visually inspecting its performance on unseen videos. Due to the low accuracy and bias towards predicting inflammation, we thus conclude the model in its current state, is too unreliable to aid a doctor in the diagnosis of patients.

To predict the separation points, two post-processing approaches were used. The Random Walker approach had a tendency to be very conservative and predict a separation point close to the end of the video. This is likely due to the walker getting 'stuck' on noise, but with noise inherently being part of the videos, it is still questionable if this approach can produce meaningful results. An evaluation on all the videos showed it on average predicted a separation point 51.85% of the total frames in the video away from the true separation point, although the variance being high, as some predictions were very accurate and others not. Based on these results, we conclude the Random Walker approach is not accurate enough to aid a doctor in the diagnosis of colitis or in the processing of endoscopy videos.

Looking at the scoring heuristic, evaluation on all the videos showed it on average predicted a separation point 51.87% of the total frames in the video away from the true separation point. It however had less variance in its predictions, as this approach was not as conservative as the Random Walker, but also not as accurate when performing well. The fact this approach provides candidate points, and at least one of these candidate points often is near the true separation point, could be beneficial for a doctor. However, because it is not always true a candidate point is close, and it being unknown which candidate point is close, it is not of much help in practice. Thus we conclude this approach is not suited for separation point prediction.

Performing treatment predictions only on the predicted segmentations from model *Idx_2_3_4_5_6_skip_20* yielded 40% accuracy, however, the predictions consisted only of classes 0 and 2. Although 40% accuracy is decent in a 5 class classification problem, due to the predictions only consisting of 2 classes, the false positive rate is high, which was also validated in the ROC curve. However, both class 0, 1 and the micro-average achieved an area under the curve a fair bit above 0.5, indicating the model to some extent is capable of distinguishing between the classes. When adding the true segmentations to the training, the accuracy increased to 48% although still only two classes were predicted, and we saw similar results in the ROC curves. Thus we conclude the treatment predictions works to some extent, but the accuracy is too low to be of aid to a doctor.

Training a U-Net to give the original segmentations some more structure, and bring them closer to the true segmentations, to some extent worked as a lot of frames classified with inflammation got overturned to healthy. However, overly many healthy frames were predicted, likely as a reaction to model *Idx_2_3_4_5_6_skip_20* overly predicting inflammation. In conclusion, the U-Net seemed reluctant to predict inflammation, but overall, the segmentations seemed to become more 'blocky' but only slightly more accurate.

To end with, treatment predictions were made on the results of the U-Net segmentations. Using both predicted and true segmentations, 20% accuracy was achieved, and class 0,1 and 2 were predicted. The ROC curve only shows class 0 having an area under the curve above 0.5, and thus it is no surprise the accuracy is low. When only training on the predicted segmentations from the U-Net, we achieve 52% accuracy while predicting classes 0,1 and 2. The ROC curve shows, however, all classes being below 0.53, which indicates although the model makes the correct predictions, it is not very confident in its predictions, and it is likely it has close to equal probabilities for each class. In conclusion 52% accuracy is fairly good on a 5 class classification task, but again, it is not reliable enough to function as a second opinion for a doctor.

8 References

[1] Healthline, https://www.healthline.com/health/colitis#types-and-causes, visited May 22 2022.

[2] LeCun et. al., 'Deep Learning', Nature Vol.521, May 2015.

[3] K. O'Shea and R. Nash, 'An Introduction to Convolutional Neural Networks', arXiv, published: 2015.

[4] Olaf Ronneberger et. al., 'U-Net: Convolutional Networksfor Biomedical Image Segmentation', Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg, Germany, Published: 2015.

[5] Kaiming He et. al., 'Deep Residual Learning for Image Recognition', Microsoft Research, December 2015.

[6] Pytorch Team, RESNET, https://pytorch.org/hub/pytorch_vision_resnet/, visited 3rd May 2022.

[7] Researchgate, 'A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer's Disease Stages Using Resting-State fMRI and Residual Neural Networks', https://www.researchgate.net/figure/Original-ResNet-18-Architecture_fig1_336642248, visited May 9th 2022.

[8] Brownlee, Jason, 'A Gentle Introduction to Cross-Entropy for Machine Learning', https://machinelearningmastery.com/cross-entropy-for-machine-learning/, visited May 9th 2022.

[9] Wikipedia, 'Gradient descent', https://en.wikipedia.org/wiki/Gradient_descent, visited May 13th 2022.

[10] D. Kingma and J. Ba, 'Adam: A method for stochastic optimization', published as a conference paper at ICLR , 2015

[11] A. Yadav, 'All about Gradient Descent and its variants', https://medium.com/analytics-vidhya/all-about-gradient-descent-and-its-variants-d095be1a833b, visited: May 13th 2022

[12] K. Voogd and S. Dahrs, 'Reproducibility project: Learning to learn by gradient descent by gradient descent', https://tudelftgroup7.medium.com/reproducibility-project-learning-to-learn-by-gradient-descent-by-gradient-descent-9fe43c3ef948, visited: May 13th 2022

[13] Srivastava, et. al., 'Dropout: A Simple Way to Prevent Neural Networks from Overfitting', Journal of Machine Learning Research 15 (2014), published June 2014.

[14] D. Vasani, 'This thing called Weight Decay', https://towardsdatascience.com/this-thing-called-weight-decay-a7cd4bcfccab, visited May 22nd 2022.

[15] LeCun et. al., 'Neural Networks: Tricks of the Trade', Springer 2nd edition, pages 16-17.

[16] M. Stewart, 'Neural Network Optimization', https://towardsdatascience.com/neural-network-optimization-7ca72d4db3e0, visited: May 29th 2022.

[17] S. Ioffe and C. Szegedy, 'Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift', ICML, 2015.

[18] S. Narkhede, 'Understanding AUC-ROC Curve', https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5, visited May 22 2022.