SuperLoss: A Superpixel-Guided Loss for Noisy Label Semantic Segmentation in X-Ray Images

Giannis Lazarou[1]

Supervisor(s): Jesse Krijthe[1]**, Gijs van Tulder**[1] **and Myrthe van den Berg**[1,2]

[1]**EEMCS, Delft University of Technology, The Netherlands**
[2]**Orthopedics and Sports Medicine, Erasmus Medical Center, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Giannis Lazarou
Final project course: CSE3000 Research Project
Thesis committee: Jesse Krijthe, Gijs van Tulder and Myrthe van den Berg, Xucong Zhang

Abstract

Deep learning based architectures have been applied to semantic segmentation tasks in medical imaging with great success. However, such models are heavily reliant on the quality of the ground truth segmentation mask and hence are susceptible to label noise. To address this issue, this paper introduces SuperLoss, a loss function that pushes semantic boundaries towards superpixel edges. Superpixels are compact, homogeneous regions within an image that group pixels with similar characteristics, such as pixel intensity. Our loss can be combined with other loss functions for different segmentation architectures. We demonstrate our framework on a combination of two large public datasets of hip joint X-Ray images. We compare a U-Net model with and without our loss, when trained with different fractions of noise in the training dataset. Our approach achieves a $1-2\%$ improvement in Intersection-over-Union and Hausdorff distance for some cases, yet yields worse in some other cases. We also perform hypothesis testing and show that our results are statistically significant with low to medium effect size.

Keywords: Semantic Segmentation, Loss Function, Superpixels, Noisy Labels, Medical Imaging

1 Introduction

Medical image segmentation is vital for various clinical tasks, including medical image recognition and registration. For example, in the context of hip joint X-rays, segmentation is useful for extracting critical anatomical structures like the femur, ischium, and foramen. This extraction can enable better analysis and assessment of hip joint health.

However, strongly supervised deep learning models in medical imaging depend on large datasets, which often suffer from label noise (Ching et al., 2018). Major sources of this noise include inter-observer variability, human annotator errors, and inaccuracies in computer-generated labels (Karimi et al., 2020). This can have significant consequences, leading to misclassifications and incorrect measurements that ultimately affect the reliability of diagnostic tools. For instance, erroneous segmentation of medical images can result in misdiagnosed conditions, leading to inappropriate or delayed treatments. Moreover, these inaccuracies can waste valuable time for medical professionals, who may need to re-evaluate and correct misclassified cases manually.

Datasets with noisy labels are, and will remain, a common occurrence in training deep learning models for medical image analysis. Therefore, developing algorithmic approaches that can effectively manage label noise is highly desirable. For the context of this paper, we should define what type of noise we are considering. Let a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ where N is the total number of image samples, x_i denotes the original medical image, and y_i is a pixel-wise semantic segmentation mask. We are only interested in noise in label y_i. That is, we consider misclassified pixel class indices in the label, despite the fact that x_i could include measurement noise.

Numerous efforts have been made to address the issue of training segmentation networks with such noisy labels, which can be broadly classified into three categories. The first category of methods is data re-weighting, which assigns different weights to training samples based on their likelihood of being correctly labeled to mitigate the impact of noisy labels on the training process. For example, Zhu et al. (2019) have proposed a method that involves training a label evaluation network alongside the segmentation network, effectively reweighting image losses by assigning weights to each training sample to down-weight those with incorrect labels. However, these image-level weighting strategies are less effective under conditions of severe noise because they fail to fully leverage the pixels with correct annotations within each image.

The second category includes refining a coarse prediction. For example, Shu et al. (2019) have proposed extracting local visual saliency regions from low level feature channels which they then utilize in a refinement module. In our case, these were difficult to approximate, as their methodology relies on Sobel filter, which under performs on our dataset. Prabakaran et al. (2023) use superpixels and post processing techniques to refine Class Activation Maps. However, they rely on image-level labels and pose their problem as a Weakly Supervised Semantic Segmentation task. Li et al. (2021) propose a learning strategy for semantic image segmentation that leverages superpixel representation and an iterative learning scheme. Their approach integrates noise-aware training with noisy label refinement, guided by superpixels, to utilize structural constraints and improve model performance despite label noise. This method extends the Co-teaching paradigm (Han et al., 2018) by jointly training two instances of the segmentation network, which is more expensive and might struggle to be integrated with other backbone segmentation networks. A general observation for these training methods is that incorporating prior image information, either in the form of edges or intensity/spatial similarities in the case of superpixels, leads to robustness against label noise.

In the third category of methods, many studies retain the model architecture, training data, and training procedures mostly unchanged, while only modifying the loss function. A notable example of this is T-loss, proposed by Gonzalez-Jimenez et al. (2023), which is based on the negative log-likelihood of the Student-t distribution, allowing for controlling sensitivity to outliers in data, based on a single parameter. This parameter is changed during backpropagation. However, they make the assumption that error terms follow a Student's t-distribution, and they do not utilize any information from the images themselves.

In this paper we propose SuperLoss, a loss function for semantic segmentation tasks, which forces coarse prediction boundaries along superpixel edges. Superpixels are perceptually meaningful and compact image regions obtained by grouping pixels with similar characteristics such as color, texture, or brightness. Our approach can be applied to any segmentation model and can be combined with other loss functions. Specifically, we examine the performance of a U-Net

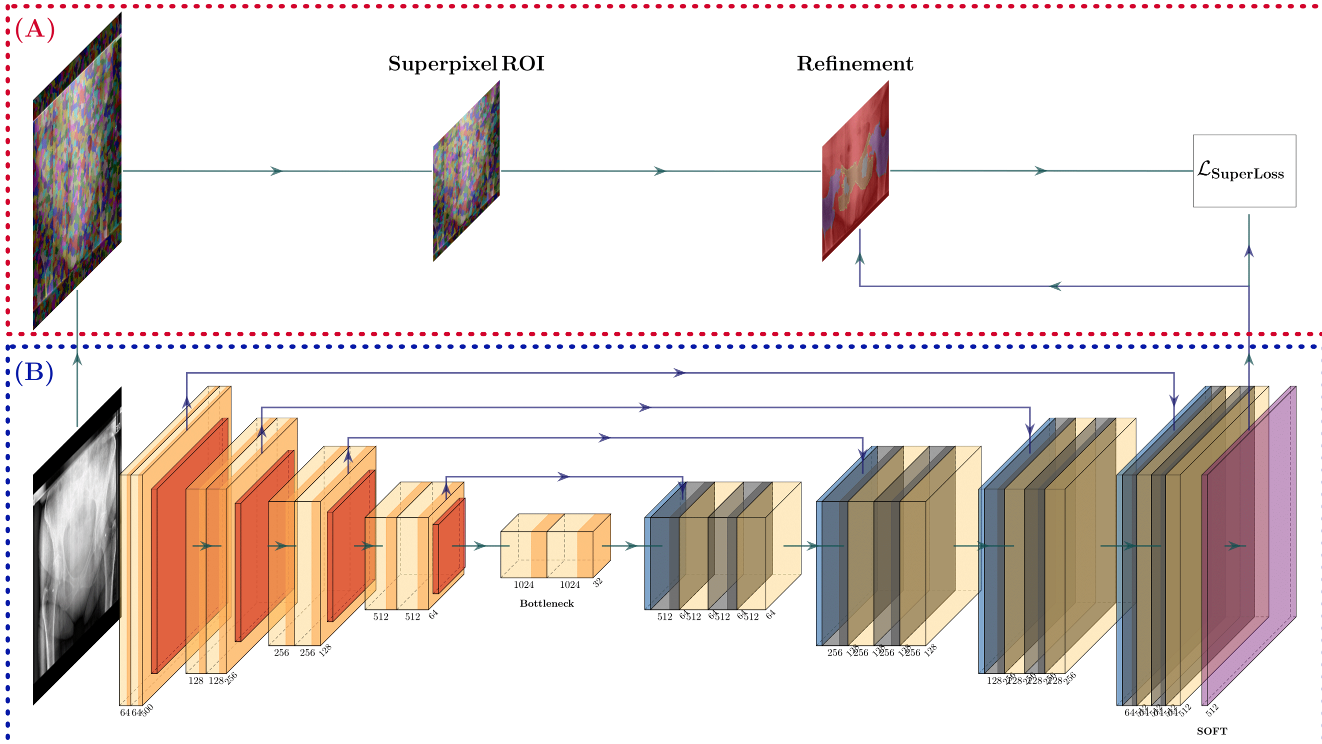

Figure 1: (A) Process of our proposed loss, and (B) a segmentation architecture

architecture (Ronneberger et al., 2015) with our additional loss function compared to one without, for femur, ischium, and foramen segmentation. We inject two distinct levels of noise into various portions of the training data. The first noise level aims to simulate typical human annotation errors whereas the second is coarser (see Section 3.1). The hypothesis is that by incorporating prior image information in the form of superpixels, our model will excel in generalizing outlier pixel misclassifications, resulting in better segmentation outcomes.

We evaluate our method on a combination of two public datasets, CHECK and OAI. Empirical results show that our loss can perform better in some cases, yet in other cases it can perform worse. We also perform hypothesis testing between the results of our baseline and our proposal and show that we achieve statistically significant differences.

The organization of this paper is as follows. In Section 2 we discuss how we generate the superpixels and how we compute our loss. In Section 3 our results will be presented, and hypothesis testing on these results is done in Section 4. In Section 5 we reflect on ethical aspects of our research. Discussion and Conclusions are presented in Sections 6 and 7 respectively.

2 Methodology

A shown in Fig. 1 our framework consists of two parts: a segmentation model and our superpixel-guided loss. For our experiments we consider a classic U-Net as our segmentation architecture due to its simple yet effective structure. Note that this basic model can also be substituted with more recent variants (Siddique et al., 2021). For instance, the attention U-Net variant (Oktay et al., 2018) integrates attention gates (Schlemper et al., 2019) into its architecture, enhancing its ability to focus on relevant features while discarding irrelevant ones.

2.1 Superpixel Generation

Superpixels are perceptually homogeneous regions within an image that approximate the underlying image structure, serving as a useful representation for segmenting and simplifying image data while preserving important boundaries and features.

Superpixel generation algorithms can generally be classified into two main types: graph-based methods and gradient ascent approaches (Ibrahim and El-kenawy, 2020). Graph-based superpixel generation methods consider each pixel as a node within a graph. The edge weights between nodes reflect the similarity between neighboring pixels. Superpixels are then formed by minimizing a cost function defined on this graph. In contrast, gradient ascent methods iteratively refine clusters until they meet a convergence criterion. We note that this is not an exhaustive list of superpixel generation algorithms, and more taxonomies have been proposed (Barcelos et al., 2024).

Deciding on an algorithm is crucial to the outcome of our approach as generating too few superpixels can result in small predictions to merge with others, and generating too many might not improve the noise enough. For the scope of this paper, we considered Felzenszwalb's (Felzenszwalb and Huttenlocher, 2004) algorithm from the graph-based category, and Quickshift (Vedaldi and Soatto, 2008), SLIC (Achanta et al., 2010), and Compact Watershed (Neubert and Protzel, 2014) from the gradient-ascent category. An example of how each algorithm performs on our dataset is given in Fig. 2.

We considered superpixels generated from a subset of images of size 5 from both OAI and CHECK, for the various hyperpameters each algorithm has. Generally, we noticed that superpixels generated with the Quickshift algorithm had the most overlap with our ground truth masks, which is what we decided to run our experiments with. SLIC outputs superpixels that are too rectangular and struggle to approximate the femur head, which is also the case with Felzenszwalb's algorithm and Compact Watershed. The hyperpameters for the Quickshift algorithm are mentioned in Section 3.1. In order to avoid a large overhead during training, we computed these superpixels beforehand.

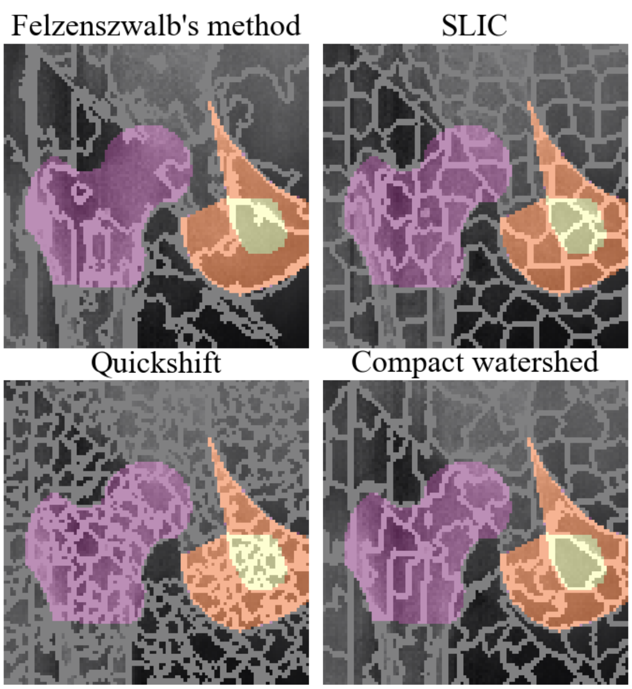

Figure 2: Comparison of different superpixel algorithms that were considered, superimposed with our ground truth mask

2.2 Refinement

A superpixel representation of a single image sample, generated from our previous step, is denoted by $s = \{s_i\}_{i=1}^{M}$, where M is the number of image pixels and $s_i \in \{1, 2, \cdots, K\}$, where K is the number of unique superpixel labels. For each $k \in \{1, 2, \cdots, K\}$ we denote $P_k = \{i \mid s_i = k\}$ as the set of pixel indices that belong to superpixel k. The output of the segmentation model is $\hat{y} = \{\hat{y}_i\}_{i=1}^{M}$ and $\hat{y}_i \in \{0, 1, \cdots, C\}$, where C represents the labels for the semantic classes. Our refinement is then defined as:

$$\tilde{y}_{i \in P_k} = \underset{c \in \{0,1,\cdots,C\}}{\arg\max} \sum_{i \in P_k} \mathbb{I}(\hat{y}_i = c), \quad \forall k \in \{1, 2, \cdots, K\} \tag{1}$$

where $\mathbb{I}(\cdot)$ is the indicator function that equals 1 if the condition inside is true, and 0 otherwise. Essentially, for every unique superpixel label we assign the most occurring semantic class from the segmentation model's output.

As applying this operation on the entire image would make our training to be much slower, we consider this refinement only around a region-of-interest (ROI). This ROI is determined per batch by identifying the first and last nonzero pixels in the mask's rows and columns, and applying an offset padding of 5 pixels to each side.

2.3 Loss function

For the baseline U-Net model, a combination of Weighted Cross Entropy and Dice Loss were used. Weighted Cross Entropy is defined as:

$$\mathcal{L}_{\text{WCE}} = -\sum_{c=0}^{C} w_c \, y_c \log(\hat{y}_c) \tag{2}$$

where C is the number of classes, y_c is the true label for the class c, $\hat{y}_c$ is the probability that the sample belongs to class c, and w_c is the weight vector. For our case we calculate w_c as the ratio of class pixels to the total image pixels. Dice Loss is defined as:

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_{i=1}^{M} y_i \hat{y}_i}{\sum_{i=1}^{M} y_i + \sum_{i=1}^{M} \hat{y}_i} \tag{3}$$

Our proposed loss is a Cross Entropy Loss between the refined mask $\tilde{y}$ and the segmentation model's output $\hat{y}$:

$$\mathcal{L}_{\text{SuperLoss}} = -\sum_{c=0}^{C} \tilde{y}_c \log(\hat{y}_c) \tag{4}$$

The final hybrid loss function is:

$$\mathcal{L} = \mathcal{L}_{\text{WCE}} + \mathcal{L}_{\text{Dice}} + \lambda \cdot \mathcal{L}_{\text{SuperLoss}} \tag{5}$$

Where λ denotes the weight coefficient of the superpixel loss. The impact of this hyperparameter is further elaborated in Section 3.4.

3 Experiment

3.1 Dataset

The dataset used for training and evaluation is a combination of both the Cohort Hip and Cohort Knee (CHECK) study (Wesseling et al., 2014), and the Osteoarthritis Initiative (OAI) study (Eckstein et al., 2014). Both of these cohorts have been under study for more than a decade, gathering longitudinal data that is typically updated annually. Although they include other clinical data, we only consider the hip-joint X-Rays from both of these studies. The datasets include multiple visits of the same participants, although the number of visits is variable, as some of the participants drop out. In total, our dataset amounts to 14858 images, where a 75-15-15 train-test-validation split was used, after shuffling all of our images.

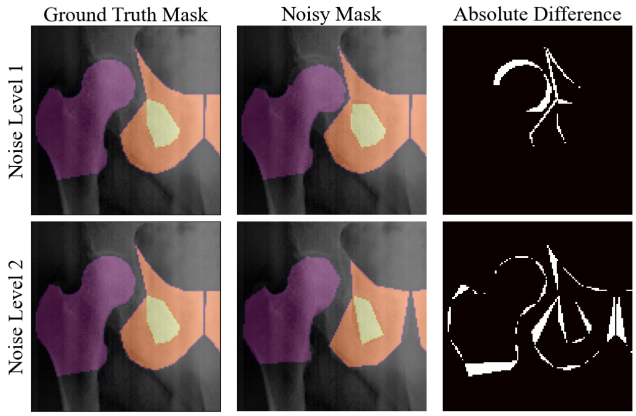

Figure 3: Cropped sections of the original mask, noisy mask, and mask differences (left to right) for noise Level 1 and Level 2 (top to bottom)

Superpixels

Our superpixels are generated with Scikit's Quickshift method with the following parameters: kernel size = 1, max distance = 2.6, and ratio = 0.7 (Pedregosa et al., 2011).

Mask Generation

For the ground truth mask without noise we connect the landmark points generated by the software BoneFinder (Lindner et al., 2013). BoneFinder is an automated software tool designed to delineate and separate skeletal structures from 2D radiographs. Although the software outputs landmark points for a variety of bone contours, we chose to focus only on femur, ischium, and foramen, as the points for these were fully connected. We connect the landmark points to form a polygon using OpenCV's fillPoly function (Itseez, 2015).

Noise Generation

For Level 1 noise, the goal was to approximate a realistic type of noise, that resembles the mistakes made by the software. A segment of $\frac{1}{3}$ the total size of each class' points was moved in a random direction by a random amount within the range $\left[0, \frac{1}{45} * \text{image width}\right]$.

For Level 2 noise, we use half the original points by interleaving every other point, to get a coarse prediction. This noise type is to test how our implementation performs in more difficult cases. Albeit not as realistic, good performance on this type of noise could potentially mean spending less time annotating, as only half the points could be used in segmentation problems. An example of both these noise types is shown in Fig. 3.

3.2 Data Preprocessing

All X-Ray image intensities were first normalized between 5th and 95th percentiles, followed by scaling the range between 0 and 1. This was to take into account the differences in intensities between the datasets. The images also differ in aspect ratio, so to preserve it we padded them with constant padding to an aspect ratio of 1:1 and then downsampled to a size of 256x256. During training we apply a random horizontal flip to the images with a probability of 0.5.

3.3 Evaluation

All our experiments were trained for 30 epochs on the DelftBlue Supercomputer's GPU nodes using a single Nvidia V100S GPU with 2 CPU cores (Delft High Performance Computing Centre , DHPC). We use Stochastic Gradient Descent with learning rate 5×10^{-2} and momentum 9×10^{-1}. We executed our code using CUDA 12, Pytorch 2.3.0, and torchvision 0.18.0. For metrics, we consider Intersection-over-Union (IoU), and Hausdorff Distance (HD), both per class and averaged. Since HD is susceptible to outlier points, we consider the 95th percentile of distances. We consider a combination of both overlap and distance metrics as they provide complementary insights into the segmentation performance. Overlap metrics like IoU give a measure of the common area between predicted and ground truth masks, indicating the overall segmentation accuracy. On the other hand, distance metrics like HD assess the spatial accuracy of boundaries, capturing how close the predicted contours are to the actual ones. Our quantitative results are summarized in Table 1 and Table 2 for Noise Level 1 and 2 respectively, for $\lambda = 1 \times 10^{-3}$. Qualitative results are shown in Fig. 4 and Fig. 6.

3.4 Hyperparameter Tuning

We evaluate the effect of hyperparameter λ from Eq. 5 by training the same model for different weight coefficients λ of our loss function. All evaluations were run with a noise fraction $\phi = 1.0$ for Noise Level 2, for 30 epochs. We use the same metrics as those in Section 3.3. The results are summarized in Table 3. Qualitatively, we plot our models prediction boundaries for these values of lambda, along with baseline and ground truth boundaries, which are shown in Fig. 5.

4 Hypothesis Testing

Comparing mean and standard deviation values from Table 1 and Table 2 alone might provide ambiguous insights, especially if the data does not follow a normal distribution. In this section, we perform pairwise statistical significance tests between the baseline's and our model's mIoU and mHD95 results, given a fixed Noise Level and fraction of noise (ϕ). Statistical significance indicates the likelihood that the observed differences between our model and the baseline are not due to chance. We report both p-values and effect sizes, as p-values show the probability of obtaining the observed results if there were no real difference, while effect sizes quantify the magnitude of the difference, providing a clearer understanding of the practical importance of our findings, which is our main research objective.

4.1 Normality Test

In order to decide whether to use a parametric or non-parametric statistical significance test, we must first check if our data follows a normal distribution. For this step we use both a statistical normality test, and a visual normality check. We chose the normality test proposed by D'Agostino and Pearson (1973), which combines skewness and kurtosis to create an omnibus test for normality.

		Noise Level 1								
ϕ	Model	IoU ↑			mIoU↑	HD95↓			mHD95↓	Train
		Fem.	Isc.	For.		Fem.	Isc.	For.		
0.0	U-Net	0.946	0.899	0.897	0.915 ±0.037	**2.301**	1.630	1.484	1.805 ±1.795	4h47m
0.0	Ours	0.946	**0.905**	**0.898**	0.917±0.037	2.347	**1.587**	**1.431**	1.789 ±1.764	18h30m
0.5	U-Net	**0.939**	0.878	0.878	0.898 ±0.035	**2.667**	**1.994**	**1.602**	2.087 ±1.811	4h53m
0.5	Ours	0.928	0.878	**0.885**	0.896 ±0.040	2.855	2.043	1.717	2.205 ±2.509	18h59m
1.0	U-Net	**0.920**	0.845	0.854	0.872 ±0.039	**3.252**	2.957	1.917	2.708 ±2.766	4h57m
1.0	Ours	0.913	**0.858**	**0.865**	0.878 ±0.036	3.431	**2.315**	**1.737**	2.494 ±1.809	18h51m

Table 1: Quantitative evaluation of our method compared to our baseline for noise Level 1 at three fractions of noise (ϕ)

		Noise Level 2								
ϕ	Model	IoU ↑			mIoU↑	HD95↓			mHD95↓	Train
		Fem.	Isc.	For.		Fem.	Isc.	For.		
0.5	U-Net	**0.928**	**0.869**	**0.852**	0.883 ±0.044	**3.576**	**2.346**	**2.042**	2.655 ±2.487	4h39m
0.5	Ours	0.926	0.856	0.840	0.874 ±0.043	4.002	2.663	2.126	2.930 ±2.547	17h52m
1.0	U-Net	0.872	0.768	0.759	0.800 ±0.029	6.932	**4.700**	**2.698**	4.777 ±1.779	4h58m
1.0	Ours	**0.873**	**0.770**	**0.762**	0.802 ±0.032	**6.842**	4.703	2.771	4.772 ±2.428	17h30m

Table 2: Quantitative evaluation of our method compared to our baseline for noise Level 2 at two fractions of noise (ϕ)

The null hypothesis for this test is H_{N0}: A sample comes from a normal distribution. Our alternative hypothesis is defined as H_{Na}: A sample does not come from a normal distribution. We chose a threshold of $a = 0.05$ before running our evaluations. Our results are summarized in Table 5, where the statistic is the squared sum of z-scores from skewness and kurtosis tests. A larger absolute value of the test statistic usually indicates a more significant deviation from the null hypothesis. Based on the fact that $p\text{-}value < a$ for all the models, we reject H_{N0} and conclude that our metrics do not follow a normal distribution.

For our visual normality checks, we make Quantile-Quantile (QQ) plots, shown in Fig. 7 and Fig. 8. A QQ plot is a graphical tool used to assess if a dataset follows a specified theoretical distribution, which in our case is the normal distribution. It plots the quantiles of the dataset against the theoretical ones; if the points (shown in blue) fall approximately along a straight line (shown in red), the data likely conforms to the specified distribution. From our results, there are significant deviations from the line, which we interpret as departures from the theoretical distribution. Thus, our conclusion that the metrics do not follow a normal distribution remain consistent with our previous statistical test.

4.2 Statistical Significance

Our data does not follow a normal distribution and our research design is within-subject (train and test sets remain the same between models). Rainio et al. (2024) propose the Wilcoxon signed-rank test (Wilcoxon, 1945) for comparing two segmentation model's metrics for such experimental setups, when the data does not follow a normal distribution. This test is used to compare two related samples, or repeated measurements on a single sample, to assess whether their population mean ranks differ. The test statistic for the Wilcoxon signed-rank test is calculated by ranking the absolute differences between paired observations, assigning signs based on the direction of the difference, and then summing the signed ranks. The effect size is calculated as:

$$r = \frac{z}{\sqrt{n}} \tag{6}$$

where z is the normalized z-statistic and n is the total number of observations (total number of samples in both groups). Effect size is a measure that quantifies the strength of the relationship between variables. Absolute values of r closer to 1 indicate strong effect compared to random noise, whereas values closer to 0 indicate little to no effect.

We define our Null Hypothesis as H_{W0}: There is no significant difference in the mIoU and mHD95 values between the baseline and our proposal. The alternative hypothesis H_{Wa} is that there is a significant difference in both mIoU and mHD95 values between the models. We chose a threshold value $a = 0.05$ to reject H_{W0}. Our results are summarized in Table 4. Since $p\text{-}value < a$ for all pairwise comparisons between our model's metrics and the baseline's, we reject H_{W0} that there is no difference between the samples, suggesting that the observed differences are unlikely to have occurred by random chance.

λ	IoU ↑			mIoU↑	HD95↓			mHD95↓
	Fem.	Isc.	For.		Fem.	Isc.	For.	
0.001	0.873	0.771	0.762	0.802 ± 0.029	6.814	4.671	2.732	4.739 ± 1.855
0.005	0.871	0.767	0.755	0.798 ± 0.031	7.001	4.758	2.827	4.862 ± 2.469
0.01	0.874	0.772	0.757	0.801 ± 0.032	6.752	4.695	2.746	4.731 ± 2.331

Table 3: Quantitative evaluation of our method for different weight coefficients λ of our loss function for noise Level 2 ($\phi = 1.0$)

Noise Level	ϕ	mIoU			mHD95		
		Statistic	p-value	$\|r\|$	Statistic	p-value	$\|r\|$
1	0.0	902363	<0.001	0.167	672163	<0.001	0.068
1	0.5	1130308	<0.001	0.055	979370	<0.001	0.127
1	1.0	785833	<0.001	0.224	1015804	<0.001	0.109
2	0.5	845926	<0.001	0.196	806295	<0.001	0.199
2	1.0	844551	<0.001	0.196	994596	<0.001	0.118

Table 4: Wilcoxon-signed rank test statistic, p-value, and effect size (r) for pairwise comparisons between the baseline's and our model's mIoU and mHD95 results

5 Responsible Research

5.1 Data

Since our research involves medical data, we adhere strictly to the guidelines outlined in the Netherlands Code of Conduct for Research Integrity (KNAW; NFU; NWO; TO2-federatie; Vereniging Hogescholen; VSNU, 2018). All data is handled with utmost security, ensuring compliance with ethical standards and patient confidentiality. Experiments are conducted on the Delft High Performance Computing Center (DHPC), instead of locally, and unnecessary sensitive information about patients is neither stored nor retained beyond what is essential for the research objectives.

We did not collect the patient's data ourselves, and adhere to the protocols of use outlined in both the OAI and CHECK datasets. We disclose that there might be sampling biases in these datasets as they include men and women of ages 45 – 79, so it is not representative of the entire population. More specifically, the focus of the studies are people at risk for symptomatic femoral-tibial knee osteoarthritis, and includes all ethnic minorities, with a focus on African-Americans.

5.2 Data Trimming

During training we left out two images that belonged to our original train set. The reason was that these X-Rays did not include the PixelSpacing attribute, and we considered the case of resampling resolutions. We include a segment of the code used for this in Fig. 9.

5.3 Use of LLMs

We disclose the use of Large Language Models (LLMs), specifically ChatGPT, during the development of our experiment as well as for writing this paper. In general, our use cases for ChatGPT can be classified in two categories: assistance with LaTeX code formatting, and brainstorming main concepts or keywords to search for. For the first category, we used it to change the formatting of our Tables. An example prompt is given in Fig. 11, and the corresponding response is given in Fig. 10. Note that we still enter our evaluation results manually. For the second category, we used it to generate keywords when conducting our literature review. An example prompt and answer is shown in Fig. 12. This approach was employed in the initial stages of the literature review, and from the identified papers, we found additional references.

6 Discussion

The main research goal of this study is evaluating the performance of a U-Net model with our loss function compared to one without, on hip joint X-Rays, when trained under various levels of label noise. Our hypothesis was that since our loss pushes boundaries towards superpixel edges, which are independent to label annotations, it should perform better than our baseline.

Quantitative findings indicate that both models performed well, with our proposal showing $\sim 1\%$ increase in IoU for certain classes when trained under greater noise fractions at both noise levels. Our proposal also demonstrated a lower HD95 for certain classes in the presence of larger noise Level 1. Specifically, it achieves an HD95 of ~ 0.6 units lower than baseline for ischium segmentation and ~ 0.2 units for foramen segmentation, when all training data comes from Level 1 noisy annotations. However, it is contradictory with our findings for noise Level 2, where the baseline performs better for most HD95 metrics. This variability in metrics could be explained by the fact that the introduced noise may not have enough overlap with the superpixels to improve the segmentation.

Interestingly, even when increasing the value of the weight coefficient λ of our function, the IoU and HD95 metrics do not appear to change drastically. Once again, this could be due to bad initialization of superpixels as they might not overlap well with our simulated noise. It could also be that we did not consider high enough weight coefficients. Moreover, the results do not change in a consistent manner. When comparing results from $\lambda = 0.001$ to $\lambda = 0.005$ the metrics become worse, yet from $\lambda = 0.005$ to $\lambda = 0.01$ there appears to be an improvement. This goes against our assumption that we would observe a steady increase/decrease.

We also report that a classic U-Net model appears to generalize well for most cases under different fractions and intensities of noise. For example there is only a 0.026 decrease in IoU for the femur class when all data is from Level 1 noise. It is also interesting that for $\phi = 0.5$ U-Net generally seems to perform better. These observations align with past research,

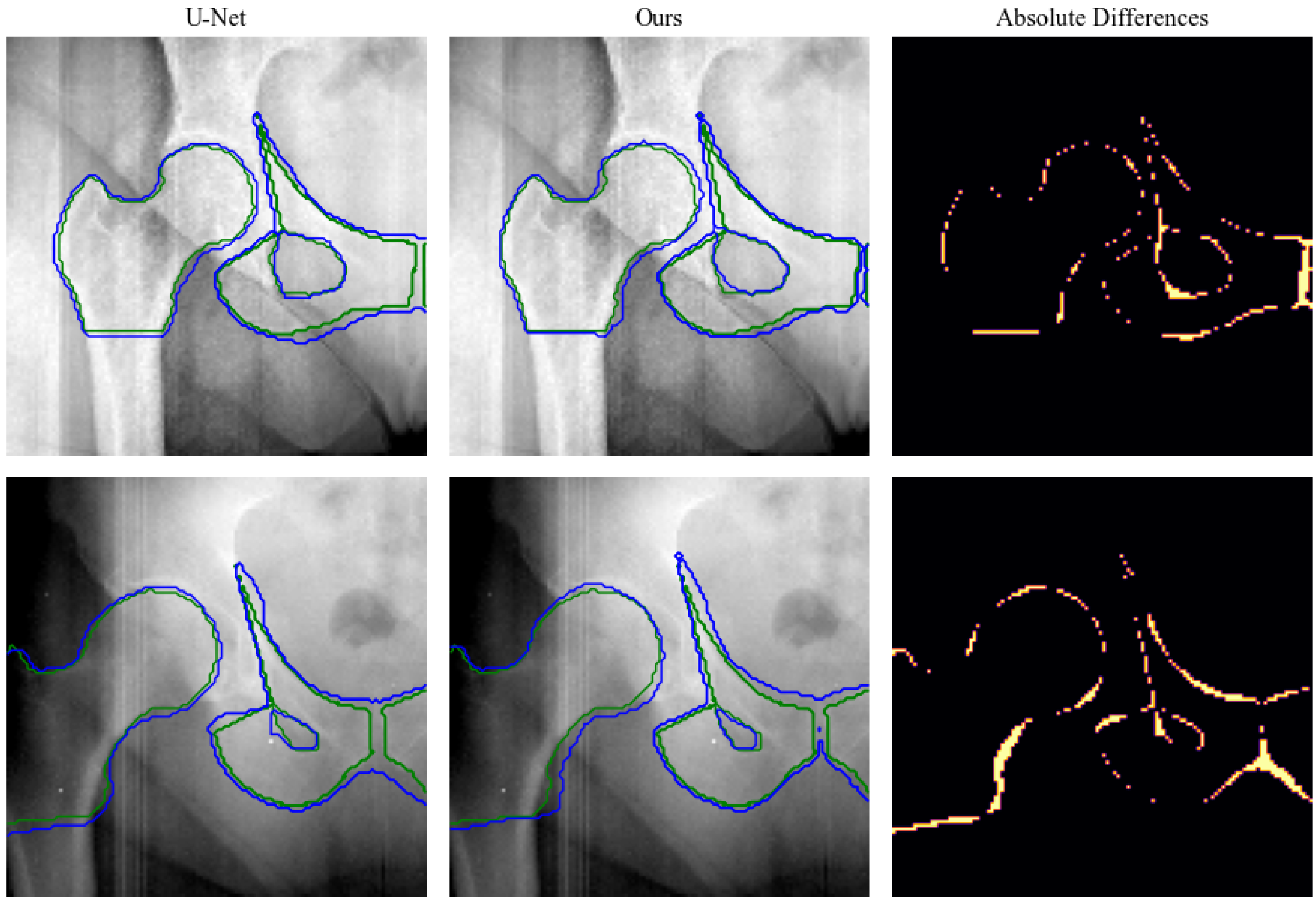

Figure 4: Qualitative results of our baseline (left) compared to our method (center), and absolute differences between them (right). Both models trained for noise Level 1 with $\phi = 1.0$. Green boundaries indicate ground truth mask, and blue indicates prediction boundaries of either model.

reinforcing the robustness and reliability of deep learning models, especially given enough training data (Zhang et al., 2016). The ability of U-Net to not show drastic decrease in performance despite varying noise levels underscores its effectiveness as a segmentation model, particularly in real-world scenarios where data quality can be unpredictable.

Based on our hypothesis testing, we achieve very small p-values when making pairwise statistical significance tests between our model's and the baseline's mIoU and mHD95 scores. Therefore, we conclude that our model has statistically significant results. The fact that our p-values are so small is expected, because when sample sizes becomes large enough, smaller differences start appearing statistically significant. From statistical significance alone, we can only infer that our proposed loss does indeed impact the outcome of both mIoU and mHD95 metrics, but nothing about the practical significance of our proposal. Effect size (r) is more relevant to our research question as it can help us quantify the degree to which our loss changes these metrics. Although to our knowledge there are no fixed cutoff values for the Wilcoxon-signed rank test's effect size, we interpret $|r| \sim 0.1$ to be a small effect and $|r| \sim 0.3$ to be a medium effect. Most of our results seem to have a small effect, which we interpret as little practical significance, especially when paired with our quantitative analysis. For certain cases however, there is an higher effect, such as for Level 1 noise at $\phi = 1.0$.

Qualitatively, we observed that even for higher weight coefficients (λ), our model did not push the boundaries as much as we had expected. In fact, it appears that our model's boundaries have quite a large overlap with the baseline's boundaries. This overlap indicates that while our proposed method has a statistically significant impact on the metrics, the practical improvement in segmentation boundaries is limited. This finding suggests that while our loss function can alter the quantitative metrics, it does not significantly enhance the qualitative aspects of boundary delineation compared to the baseline, for noise Level 2. For noise Level 1, there seems to be a bigger difference qualitatively.

Overall, while one model may not universally outperform the other, our results indicate that incorporating our loss function can lead to statistically significant improvements with small to medium effect sizes. The benefit of our loss function may not be immediately evident in all scenarios but can offer tangible improvements in specific contexts, especially

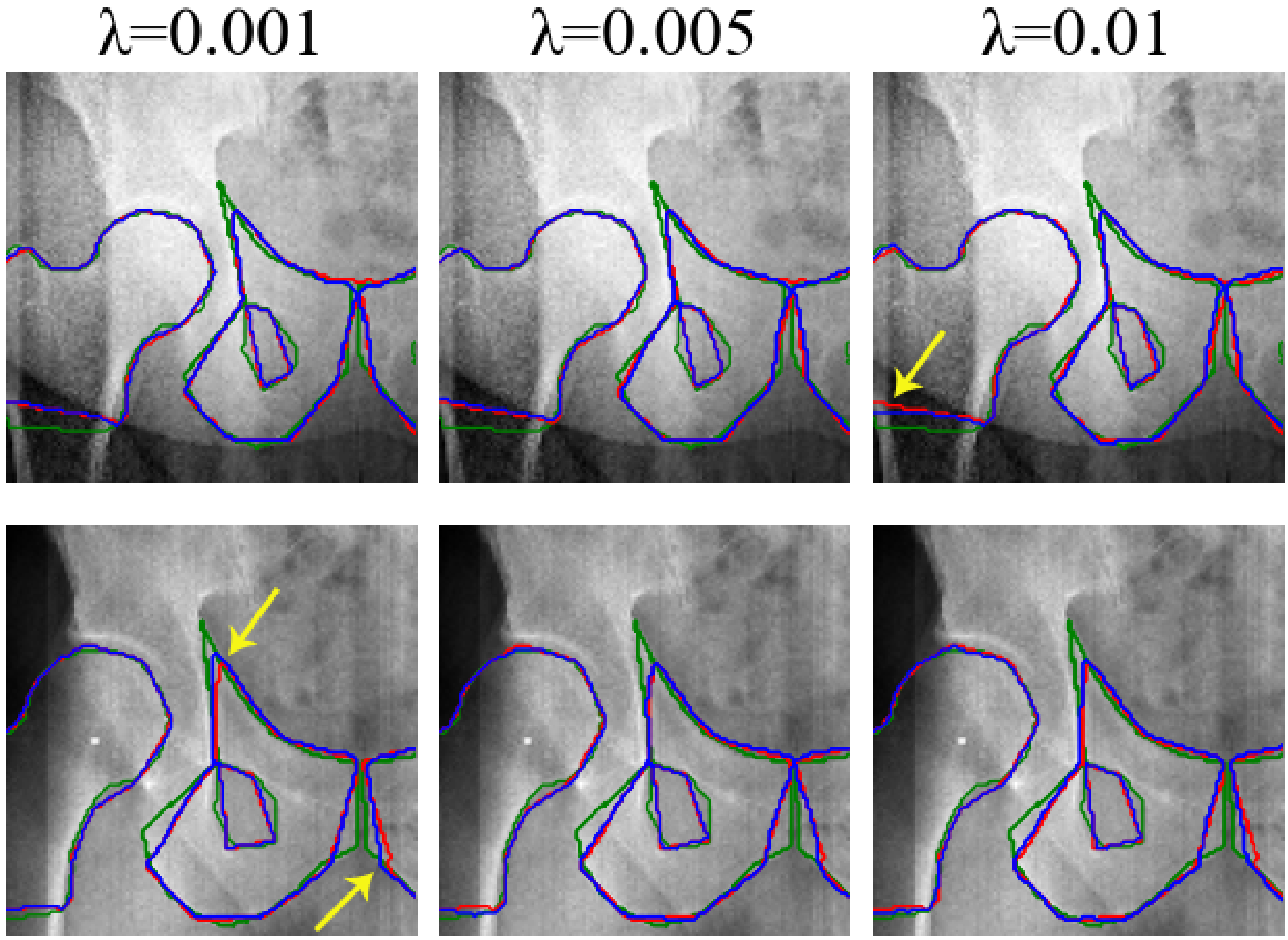

Figure 5: Qualitative results on two images of the test set, overlayed with ground truth boundaries (shown in green), baseline model's prediction boundaries (shown in red), and our model's prediction (shown in blue) across various weight coefficients λ, trained with Level 2 noise for $\phi = 1.0$.

with proper superpixel initialization.

7 Conclusions and Future Work

This paper introduces SuperLoss, a loss function for training segmentation models when a fraction of the data includes noisy labels. It can easily be used in conjunction with other segmentation losses. We exploit pixel-level information from a superpixel representation of an image, and push model's predictions towards the edges of those superpixels.

Our research question was to evaluate the performance of SuperLoss on a classic U-Net model, compared to one without. For our experiments, we first train both these models under different noise intensities and fractions of noise in the training data, then evaluate on clean annotations. We considered mean Intersection-over-Union (IoU) and Hausdorff Distance at 95th percentile (HD95) as our metrics, and perform statistical significance tests on each pair of models.

Our results show a $1 - 2\%$ improvement in IoU and HD95 of some classes for certain cases. However, in some other cases our approach can be worse than the baseline. These differences are statistically significant, yet the effect size is relatively small. Qualitatively, our loss does not seem to push boundaries as much as we had expected.

Our work leaves many openings for further research. The current implementation relies on sequentially looping through unique superpixel labels to find the most occurring semantic class, which adds a large overhead to training times, as we do not take advantage of vectorization. Furthermore, our loss is calculated in a naive manner. It would be worth investigating whether weighting the loss of each superpixel separately, depending on how much the majority class overlapped compared to other classes, improves our approach. The outcome of our loss is also quite dependent on the initialization of superpixels, which involves tuning many hyperparameters.

Appendices

A Qualitative results

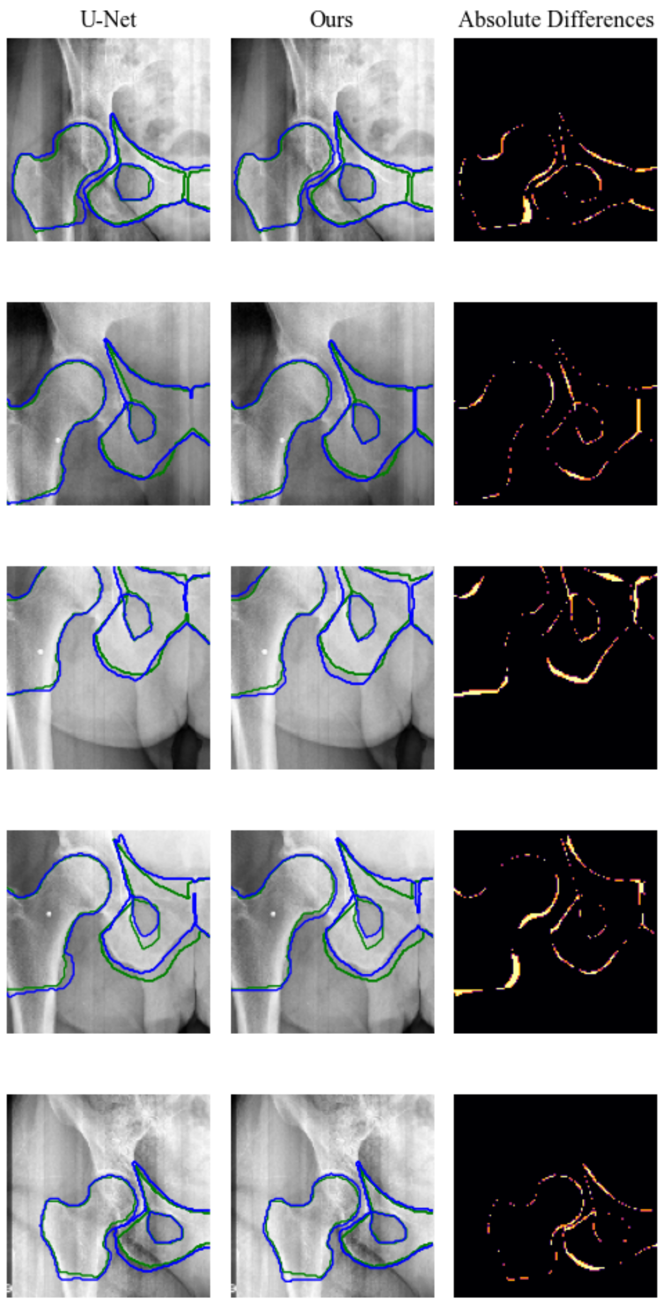

Figure 6: Qualitative results of our baseline (left) compared to our method (center), and absolute differences between them (right). Both models trained for noise Level 1 with $\phi = 1.0$. Green boundaries indicate ground truth mask, and blue indicates prediction boundaries of either model.

B Statistical Significance Supplementary

B.1 Q-Q Plots

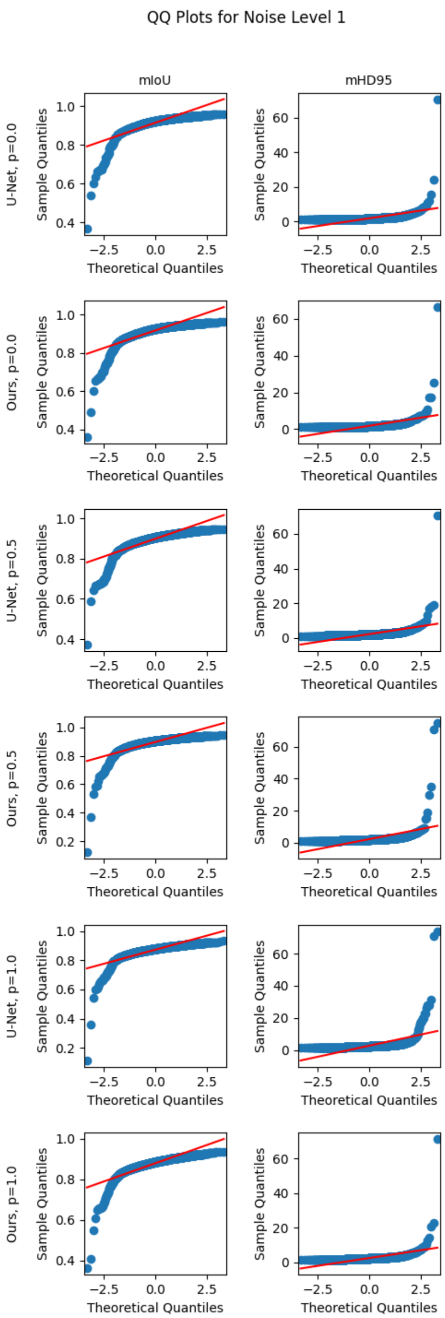

Figure 7: Quantile-Quantile plots for mIoU and mHd95 scores of all models trained under Noise Level 1. Blue line indicates our samples' quantiles, and red line indicates the theoretical quantiles that a normal distribution would follow.

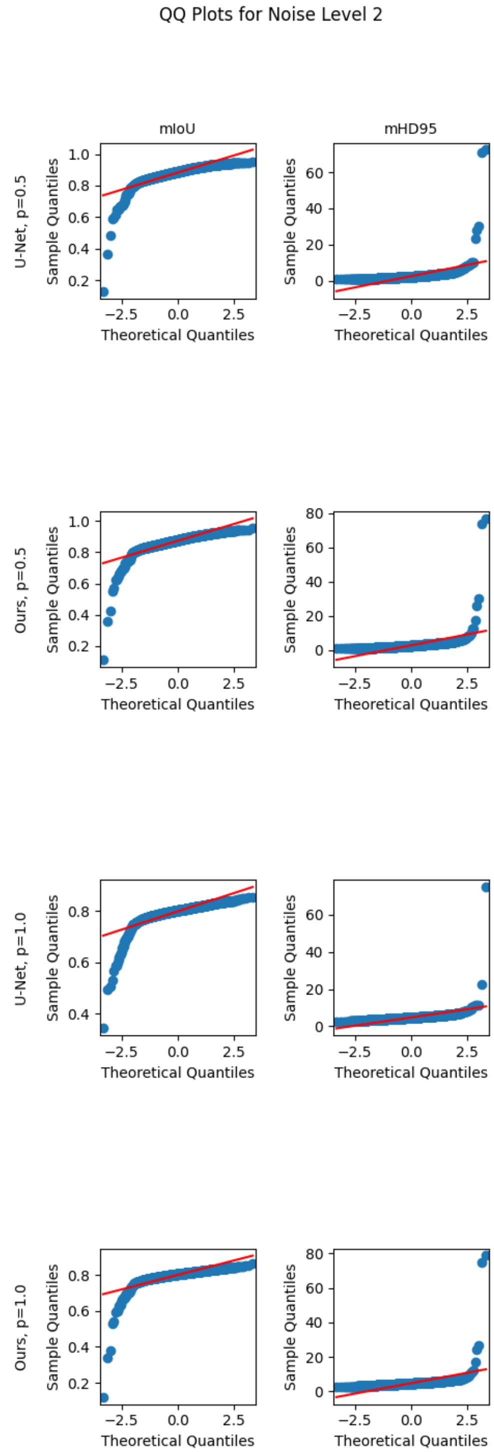

Figure 8: Quantile-Quantile plots for mIoU and mHd95 scores of all models trained under Noise Level 2. Blue line indicates our samples' quantiles, and red line indicates the theoretical quantiles that a normal distribution would follow.

B.2 Normality Test Results

Noise Level	ϕ	Model	mIoU		mHD95	
			Statistic	p-value	Statistic	p-value
1	0.0	U-Net	2240.711	<0.001	5634.228	<0.001
1	0.0	Ours	2279.620	<0.001	5377.243	<0.001
1	0.5	U-Net	2078.108	<0.001	5540.786	<0.001
1	0.5	Ours	2181.542	<0.001	5218.351	<0.001
1	1.0	U-Net	2050.636	<0.001	4463.771	<0.001
1	1.0	Ours	2236.445	<0.001	5575.988	<0.001
2	0.5	U-Net	2317.545	<0.001	5021.539	<0.001
2	0.5	Ours	2392.020	<0.001	5135.970	<0.001
2	1.0	U-Net	2404.224	<0.001	5728.291	<0.001
2	1.0	Ours	2535.091	<0.001	5521.908	<0.001

Table 5: D'Agostino and Pearson Normality test results for the metrics our model and our baseline achieve, for different noise levels and fractions of noise (ϕ)

C Responsible Research Supplementary

```python
import numpy as np
import pydicom

def load_dicom_image(dicom_path):
    # load the DICOM image and apply the PhotometricInterpretation header
    # (if necessary)

    img = pydicom.dcmread(dicom_path)

    pixel_spacing = img.get('PixelSpacing') or img.get('ImagerPixelSpacing')
    assert pixel_spacing is not None, 'no pixel spacing found'
    assert pixel_spacing[0] == pixel_spacing[1], \
            'anisotropic pixel spacing is untested'

    pixels = img.pixel_array
```

Figure 9: Code which was used to find DICOM images that did not include the attribute of 'PixelSpacing'

Sure, here's the updated LaTeX table with "Level 1" added to every row under the "Noise" column, and with "p", "Noise", and "Accuracy" centered:

latex Copy code

```latex
\renewcommand{\tabcolsep}{0.11cm}
\begin{table} [h]
    \centering
    \begin{tabular}{|c|c|c|c|c|c|c|}
        \hline
        \textbf{Noise} & \textbf{p} & \multicolumn{4}{|c|}{\textbf{Class IoU}} & \textbf{A
        \cline{3-6}
        & & \textbf{Background} & \textbf{Femur} & \textbf{Ischium} & \textbf{Foramen} & \
        \hline
        \textbf{Level 1} & \textbf{0.0} & 0.9931 & 0.9506 & 0.9207 & 0.9080 & \textbf{0.97
        \textbf{Level 1} & \textbf{0.1} & 0.9930 & 0.9489 & 0.9204 & 0.9072 & \textbf{0.97
        \textbf{Level 1} & \textbf{0.2} & 0.9927 & 0.9480 & 0.9167 & 0.9050 & \textbf{0.97
        \textbf{Level 1} & \textbf{0.3} & 0.9928 & 0.9485 & 0.9173 & 0.9047 & \textbf{0.97
        \textbf{Level 1} & \textbf{0.4} & 0.9923 & 0.9468 & 0.9119 & 0.9018 & \textbf{0.97
        \textbf{Level 1} & \textbf{0.5} & 0.9923 & 0.9446 & 0.9145 & 0.9022 & \textbf{0.97
        \textbf{Level 1} & \textbf{1.0} & 0.9913 & 0.9396 & 0.9053 & 0.8930 & \textbf{0.96
        \hline
    \end{tabular}
    \captionsetup{justification=centering}
    \caption{Class IoU scores and mean Accuracy for different fractions of noise in the tr
    \label{tab:first_perm}
\end{table}
```

Figure 10: ChatGPT's response to asking it to change the formatting of a table

can you add Level 1 in every row under Noise column and make the table tigher? Also center p, noise, and accuracy.

```latex
\renewcommand{\tabcolsep}{0.11cm}
\begin{table} [h]
  \centering
  \begin{tabular}{|c|c|c|c|c|c|c|}
    \hline
    \textbf{Noise} & \textbf{p} & \multicolumn{4}{|c|}{\textbf{Class
IoU}} & \textbf{Accuracy} \\
    \cline{3-6}
    & & \textbf{Background} & \textbf{Femur} & \textbf{Ischium} &
\textbf{Foramen} & \\
    \hline
    Level 1 & 0.0 & 0.9931 & 0.9506 & 0.9207 & 0.9080 & 0.9739 \\
    0.1 & 0.1 & 0.9930 & 0.9489 & 0.9204 & 0.9072 & 0.9739 \\
    0.2 & 0.2 & 0.9927 & 0.9480 & 0.9167 & 0.9050 & 0.9746 \\
    0.3 & 0.3 & 0.9928 & 0.9485 & 0.9173 & 0.9047 & 0.9733 \\
    0.4 & 0.4 & 0.9923 & 0.9468 & 0.9119 & 0.9018 & 0.9724 \\
    0.5 & 0.5 & 0.9923 & 0.9446 & 0.9145 & 0.9022 & 0.9724 \\
    1.0 & 1.0 & 0.9913 & 0.9396 & 0.9053 & 0.8930 & 0.9694 \\
    \hline
  \end{tabular}
  \captionsetup{justification=centering}
  \caption{Class IoU scores and mean Accuracy for different fractions
of noise in the training data}
  \label{tab:first_perm}
\end{table}
```

Figure 11: Prompt for ChatGPT to change the formatting of a table

What are some keywords I can use to conduct literature review for a segmentation task with noisy labels?

When conducting a literature review for a segmentation task with noisy labels, you can use the following keywords and phrases to search for relevant studies:

1. Noisy labels
2. Weakly supervised segmentation
3. Label noise in segmentation
4. Robust segmentation methods
5. Error-robust segmentation
6. Learning with noisy annotations
7. Noise-tolerant segmentation
8. Label correction in segmentation
9. Uncertainty in segmentation
10. Crowd-sourced annotation for segmentation

These keywords should help you identify relevant research that addresses the challenges and methods associated with segmentation tasks using data with noisy or imperfect annotations.

Figure 12: Prompt and response for keyword generation using ChatGPT

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2010). Slic superpixels. *Technical report, EPFL.*

Barcelos, I. B., Belém, F. D. C., João, L. D. M., Patrocínio, Z. K. G. D., Falcão, A. X., and Guimarães, S. J. F. (2024). A comprehensive review and new taxonomy on superpixel segmentation. *ACM Comput. Surv.*, 56(8).

Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., Kalinin, A. A., Do, B. T., Way, G. P., Ferrero, E., Agapow, P.-M., Zietz, M., Hoffman, M. M., Xie, W., Rosen, G. L., Lengerich, B. J., Israeli, J., Lanchantin, J., Woloszynek, S., Carpenter, A. E., Shrikumar, A., Xu, J., Cofer, E. M., Lavender, C. A., Turaga, S. C., Alexandari, A. M., Lu, Z., Harris, D. J., DeCaprio, D., Qi, Y., Kundaje, A., Peng, Y., Wiley, L. K., Segler, M. H. S., Boca, S. M., Swamidass, S. J., Huang, A., Gitter, A., and Greene, C. S. (2018). Opportunities and obstacles for deep learning in biology and medicine. *Journal of the Royal Society Interface*, 15(141):20170387. © 2018 The Authors.

D'Agostino, R. and Pearson, E. S. (1973). Tests for departure from normality. empirical results for the distributions of b2 and b1. *Biometrika*, 60(3):613–622.

Delft High Performance Computing Centre (DHPC) (2024). DelftBlue Supercomputer (Phase 2). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2.

Eckstein, F., Kwoh, C. K., Link, T. M., and (2014). Imaging research results from the osteoarthritis initiative (oai): a review and lessons learned 10 years after start of enrolment. *Annals of the Rheumatic Diseases*, 73(7):1289–1300.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181.

Gonzalez-Jimenez, A., Lionetti, S., Gottfrois, P., Gröger, F., Pouly, M., and Navarini, A. (2023). Robust t-loss for medical image segmentation.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I. W., and Sugiyama, M. (2018). Co-sampling: Training robust networks for extremely noisy supervision. *CoRR*, abs/1804.06872.

Ibrahim, A. and El-kenawy, E.-S. (2020). Image segmentation methods based on superpixel techniques: A survey. 1:1–10.

Itseez (2015). Open source computer vision library. https://github.com/itseez/opencv.

Karimi, D., Dou, H., Warfield, S. K., and Gholipour, A. (2020). Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. *Medical Image Analysis*, 65:101759.

KNAW; NFU; NWO; TO2-federatie; Vereniging Hogescholen; VSNU (2018). Nederlandse gedragscode wetenschappelijke integriteit.

Li, S., Gao, Z., and He, X. (2021). Superpixel-guided iterative learning from noisy labels for medical image segmentation. *CoRR*, abs/2107.10100.

Lindner, C., Thiagarajah, S., Wilkinson, J. M., arcOGEN Consortium, T., Wallis, G. A., and Cootes, T. F. (2013). Fully automatic segmentation of the proximal femur using random forest regression voting. *IEEE Transactions on Medical Imaging*, 32(8):1462–1472.

Neubert, P. and Protzel, P. (2014). Compact watershed and preemptive slic: On improving trade-offs of superpixel segmentation algorithms. pages 996–1001.

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M. C. H., Heinrich, M. P., Misawa, K., Mori, K., McDonagh, S. G., Hammerla, N. Y., Kainz, B., Glocker, B., and Rueckert, D. (2018). Attention u-net: Learning where to look for the pancreas. *CoRR*, abs/1804.03999.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Prabakaran, B. S., Ostrowski, E., and Shafique, M. (2023). Refit: A framework for refinement of weakly supervised semantic segmentation using object border fitting for medical images.

Rainio, O., Teuho, J., and Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, 14(1):6086.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham. Springer International Publishing.

Schlemper, J., Oktay, O., Schaap, M., Heinrich, M., Kainz, B., Glocker, B., and Rueckert, D. (2019). Attention gated networks: Learning to leverage salient regions in medical images. *Medical Image Analysis*, 53:197–207.

Shu, Y., Wu, X., and Li, W. (2019). Lvc-net: Medical image segmentation with noisy label based on local visual cues. In Shen, D., Liu, T., Peters, T. M., Staib, L. H., Essert, C., Zhou, S., Yap, P.-T., and Khan, A., editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pages 558–566, Cham. Springer International Publishing.

Siddique, N., Paheding, S., Elkin, C. P., and Devabhaktuni, V. (2021). U-net and its variants for medical image segmentation: A review of theory and applications. *IEEE Access*, 9:82031–82057.

Vedaldi, A. and Soatto, S. (2008). Quick shift and kernel methods for mode seeking. In Forsyth, D., Torr, P., and Zisserman, A., editors, *Computer Vision – ECCV 2008*, pages 705–718, Berlin, Heidelberg. Springer Berlin Heidelberg.

Wesseling, J., Boers, M., Viergever, M. A., Hilberdink, W. K., Lafeber, F. P., Dekker, J., and Bijlsma, J. W. (2014). Cohort Profile: Cohort Hip and Cohort Knee (CHECK) study. *International Journal of Epidemiology*, 45(1):36–44.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530.

Zhu, H., Shi, J., and Wu, J. (2019). Pick-and-learn: Automatic quality evaluation for noisy-labeled image segmentation. *CoRR*, abs/1907.11835.