university of groningen / Faculty of Science and Engineering / Biomedical Engineering

Improving Computer Tomography COVID-19 Lesion Segmentation Using Reproducible Pathways and Data Augmentation Techniques on Multi-demographic Cohorts

Bachelors in Artificial intelligence

University of Groningen

Bogdan Palfi
Student number: S3984451

Supervisor: Yeshaswini Nagaraj, MSc
Project lead: Dr. Peter Van Oojien

Period: 01/07/2021 - 15/05/2022

Contents

1 Introduction **4**
1.1 Aim of this work 5
1.2 Outline of the thesis 5

2 Methodology **6**
2.1 Data Description 6
2.2 Deep learning models 6
2.2.1 U-Net 6
2.2.2 Dynamic U-Net (nnU-Net) 7
2.3 Reproducible techniques 8
2.3.1 Lung segmentation 8
2.3.2 Data augmentation 8
2.3.3 Common evaluation metrics 9
2.4 Evaluation of the models 10
2.4.1 Training and model parameters 10
2.4.2 Data Split 10

3 Results **12**

4 Discussion **16**
4.1 Limitations 17

5 Conclusion and future work **17**

6 Ethics **19**

7 Acknowledgement **20**

8 Additional Contributions **20**

A Appendix **24**

B Appendix - Poster **28**

Abstract

The COVID-19 pandemic has put an immense pressure on healthcare systems around the world. To release some of this pressure, scientists attempted to include machine learning techniques in the infection testing process. This was done with the help of medical imaging and was intended as a solution to the lack of fast, reliable testing capabilities. However, reproducing and comparing different possible solutions is difficult since there is no baseline open-source model or data to compare the results with. For this reason, the current thesis aims to evaluate data augmentation techniques to improve automatic COVID-19 lesion segmentation by using reproducible techniques. These techniques come in the form of public data and open-source frameworks for preprocessing images and for creating the models. One such open-source framework is MONAI, a "freely available, community-supported, PyTorch-based framework for deep learning in healthcare imaging" [1]. Two deep learning models, U-Net and Dynamic U-Net (nnU-Net), implemented using MONAI, were employed to segment COVID-19 lesion. These models were trained and tested on 329 computed tomography scans which were publicly available. In order to improve the performance of the two models, data augmentation techniques in the form of intensity projections, noise filters and sliding windows were used. Overall, regardless of which model was tested, using sliding windows and not adding any noise lead to the most optimal results (AUC = 0.81 or AUC = 0.77, depending on which testing dataset was used). Intensity projection did not seem to improve the results in any situation. The goal of this thesis was to use open-source data and frameworks to improve lesion segmentation models. However, this approach presents certain challenges in the form of limited, low quality data. Therefore, the aim for future research would be to increase the amount of high-quality data with the help of professional radiologists in order to avoid some of the limitations present in this thesis.

Keywords: COVID-19, lesion segmentation, U-Net, Dynamic U-Net

1 Introduction

The COVID-19 pandemic has had a massive economic and social impact on the entire world. For this reason, research on the SARS-CoV-2 coronavirus has benefited from ample funding from private and public investors. A substantial part of this research has focused on how Machine Learning (ML) can help determine whether a person has contracted the coronavirus and also conduct severity assessment [2].

For these feats to be possible, diagnostic imaging techniques such as computed tomography (CT) and chest X-Ray (CXR) were employed to analyze the patients' lungs. However, research has shown that CT seems to be more sensitive to COVID-19 than CXR, primary because 3D CT scans can represent the lung more accurately [3, 4]. Furthermore, [5] found that using CT scans in conjunction with convolutional neural network (CNN) models such as Resnet-18, InceptionV3 and MobileNetV2 leads to better performance when detecting positive COVID-19 cases. One explanation would be that CT scans provide a better contrast and more detailed images than CXR [5]. For these reasons, the current thesis will use CT scans for its analyses, rather than CXR. Comparing CT with the current standard of Real-Time PCR (RT-PCR), CT scans were shown to have a higher sensitivity but lower specificity [6, 7]. On the other hand, RT-PCR implies a low sensitivity and high processing time but a higher specificity. Therefore, it was argued that CT be used for confirmed or clinically suspected patients with COVID-19 in order to provide circumstantial evidence [6, 7]. This would be applicable especially in situations where RT-PCR resources such as testing kits, laboratory personnel or time were limited. It is worth mentioning that RT-PCR tests can be influenced by aspects such as the reliability of the sampling kit or testing site [7]. This should prove as an additional reason for including CT scans in the analysis of confirmed or suspected patients.

Despite the advantages CT scans provide, the large number of COVID-19 infections leads to a vast amount of CT images that need to be analyzed by professional radiologists in order to reach a diagnosis. This analysis consists of two steps: correctly detecting and highlighting lung lesions, known as lesion segmentation; and deciding whether the found opacities were indeed caused by the suspected disease, step known as classification [8]. Both steps can be time-consuming and require professional radiologists, both of which can be hard to find in the context of a global pandemic. One solution to this challenge would be to use machine learning methods such as deep-learning in order to speed up the segmentation and classification steps by automatizing them. In this way, there would be less subjectivity in diagnostics and radiologists would have more time to focus on patient care and treatment. However, this approach is imperfect since it can be less accurate than actual radiologists, which is unsatisfactory, especially in the medical field. For context, deep learning often functions under the hypothesis that the training and testing data are selected from the same distribution [9], which is unfeasible in real-world scenarios.

All in all, the reviewed literature seems to present a promising future for automated diagnostic imaging, since it includes ML models with high performance when it comes to lesion segmentation in COVID-19 patients. This can be observed in table 1 as well as in the review papers [2] and [10]. However, the current literature has a considerable limitation in the sense that there are no baseline datasets nor any standard metrics when training and testing the ML models. Therefore, model comparison becomes a difficult task since there

Paper	Model	Metric	Data Augmentation
[11]	COPLE-Net	0.8 ± 0.11 Dice	Random Gaussian Noise
[12]	ResNet (COLI-Net)	0.91 ± 0.038 Dice	Intensity Clipping
[13]	semi-CARes-UNet	0.77 Dice	No data augmentation
[14]	DeepSC-COVID	0.73 ± 0.08 Dice	CT Truncation
[10]	U-Net + attention mechanism	0.69 Dice	No data augmentation
[10]	3D U-Net (MONAI)	0.87 AUC	Noise generation, scaling, rotation
[10]	U-Net	0.83 Dice	Rotation, Flip, Translation, Cropping
[10]	3D multi-decoder VNet	0.77 Dice	No data augmentation
[10]	U-Net	0.88 Dice	No data augmentation

Table 1: Overview of deep-learning models tasked with COVID-19 lesion segmentation, their results and possible data augmentation techniques employed

is no standardized procedure for creating and analyzing different models. This is especially troublesome in the medical field, which is more demanding because people's lives are at stake. Therefore, future research should aim for using standardized procedures so that improvements in the performance of the models can be more easily recognized.

For this reason, the current thesis aims to improve the classic CT COVID-19 lesion segmentation step by using reproducible pathways and data augmentation techniques. Lesion segmentation in itself is a difficult task because lesions are heterogeneous. Moreover, they can be in proximity to blood vessels or the pleura and can also present cavitary cases, all of which make segmentation even harder [8]. However, disease-specific augmentation techniques such as intensity projections, noise filters and other heuristic or non-heuristic data augmentation techniques might be able to overcome these difficulties, leading to a model generalizable in terms of multiple source data. Such a model could further be deployed and used in clinical decision-making.

1.1 Aim of this work

The main focus of this thesis is the analysis and validation of multiple data augmentation techniques and reproducible frameworks to improve the segmentation performance and reliability of two neural network models: U-Net and Dynamic U-Net (also known as nnU-Net). The main reasoning behind the choice of these methods is that they might prove effective at increasing the generalization potential of the two models when it comes to unseen domains. The models will be trained, validated and tested on public datasets containing lung CT scans of COVID-19 infected individuals.

1.2 Outline of the thesis

The thesis will first present a short description about the used datasets and introduce the architectures of the deep learning models that will be analyzed. Afterwards, the reproducible techniques of data preprocessing and augmentation, as well as the evaluation metrics used will be covered. The hyperparameter variations will then be explained, along with the data split. and ending with the results and discussions. Finally, the results of the two models will be presented and discussed.

Dataset	# of scans	Type	Previously labeled
COVID-19-20 Grand Challenge	250	CT	Yes
Moscow	50	CT	Yes
Coronacases Initiative and Radiopedia	29	CT	Yes

Table 2: Overview of the datasets used for training and testing. All datasets contained lung CT scans which presented lesions previously labeled by professional radiologists.

2 Methodology

2.1 Data Description

In total, 3 different public datasets were used for training and testing the two models, summing up to 329 lung CT scans (see Table 2). The main dataset was used for the COVID-19-20 Grand Challenge and contains unenhanced chest CTs from 250 patients that have a positive RT-PCR test for SARS-Cov-2. "The annotation of the dataset was made possible through the joint work of Children's National Hospital, NVIDIA and National Institutes of Health for the COVID-19-20 Lung CT Lesion Segmentation Grand Challenge" [15, 16, 17]. Furthermore, 50 anonymized lung CT scans from the municipal hospitals in MOSCOW were included. The scans were previously labeled by experts of Research and Practical Clinical Center for Diagnostics and Telemedicine Technologies of the Moscow Health Care Department [18]. Finally, 29 CT scans from the Coronacases Initiative (10 CTs) and Radiopedia (19 CTs) were also used, having previously being labeled by two radiologists and verified by a third experienced radiologist [19, 20].

2.2 Deep learning models

The two deep learning models that will be analyzed are U-Net [21] and Dynamic U-Net (nnU-Net) [22], implemented in the MONAI framework [1].

2.2.1 U-Net

U-Net is a CNN architecture, formed out of 23 layers and specialized in semantic segmentation. This architecture can also be observed in Figure 1. Its functioning is given by two paths, a contracting path (i.e. the encoder), tasked with capturing the context of the input, followed by an expansive path (i.e. the decoder), used for precise localization. The contracting path is represented by consecutive applications of convolution blocks followed by a 2×2 max pooling operation with stride 2 which ensures downsampling. The convolution blocks are made out of two 3×3 unpadded convolutions ending in a rectified linear unit (ReLU) [21]. The number of feature channels is also doubled every time downsampling is performed. The expansive path is symmetric to the contracting one, starting by upsampling the feature map then halving the number of feature channels using a 2×2 up-convolution. The result of this step is concatenated with the corresponding cropped feature map extracted in the contracting part. Finally, the same convolution blocks are applied (i.e. two 3×3 convolutions followed by a ReLU) [21]. At the end, an 1×1 convolution maps the created feature vectors to each class.

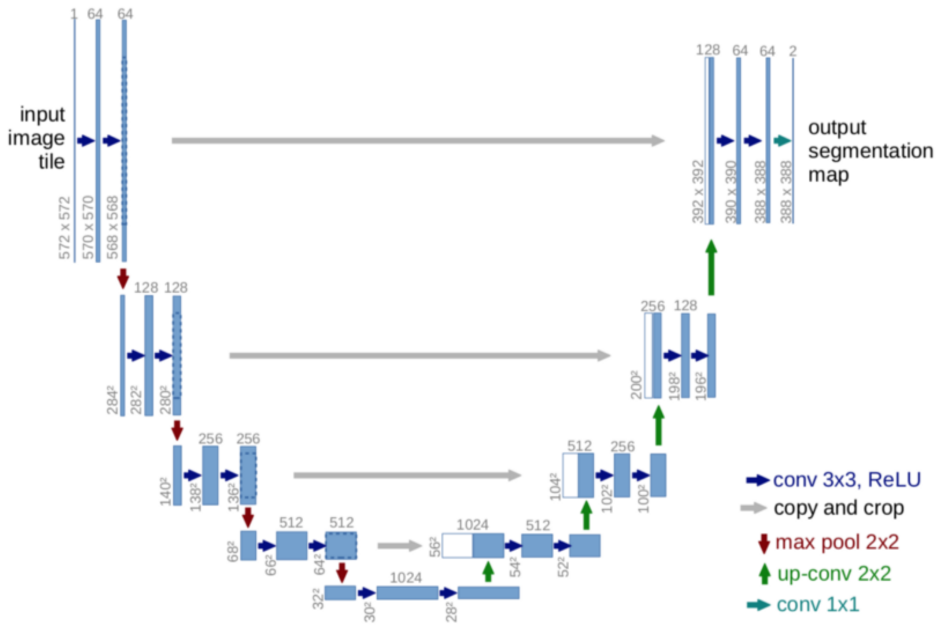


Figure 1: The architecture of U-Net. The input image is passed to the contracting path (left side of the U shape), which captures the context of the image. The output of this path along with the corresponding cropped feature map (see horizontal lines) are then passed to the expansive path (right side of the U shape). This path is used for precise localization. The output represents a segmentation map of the input image. [21]

Hyperparameters : U-Net was trained using the Adam optimizer with a learning rate of 10^{-3}, a dropout of 0.5 and a batch size of 6. For the learning rate, the Reduce Learning Rate on Plateau scheduler was used to fine tune the initial value. The loss function represented a weighted sum of the Dice loss and Cross entropy loss, each accounting for 50% of the sum. Finally, the maximum number of epochs was set to 500 in the context of early stopping with a patience of 30 and a required improvement for each 30 epochs of at least 0.01.

2.2.2 Dynamic U-Net (nnU-Net)

Dynamic U-Net , initially named nnU-Net [22], is an end-to-end automated pipeline (see Figure 2), specifically created for medical image segmentation. This pipeline starts by inferring the data fingerprint through heuristic rules. This fingerprint contains data-dependent hyperparameters such as the image sizes and spacings, as well as modalities, the number of classes and the number of training cases. The fingerprint is then merged with the inferred and blueprint parameters to generate pipeline fingerprints, which create network training for 2D, 3D and 3D-Cascade U-Net [22]. The blueprint parameters mainly represent the architecture, faithfully resembling the original U-Net, the training schedule and parameters as well as the inference procedure. As for the inferred parameters, they constitute the image resampling and normalization, together with the batch and patch sizes. Finally, an ensemble of network configurations combined with post-processing finds the best average Dice coefficient for the training set, leaving the optimal configuration for predictions on the test set [22].

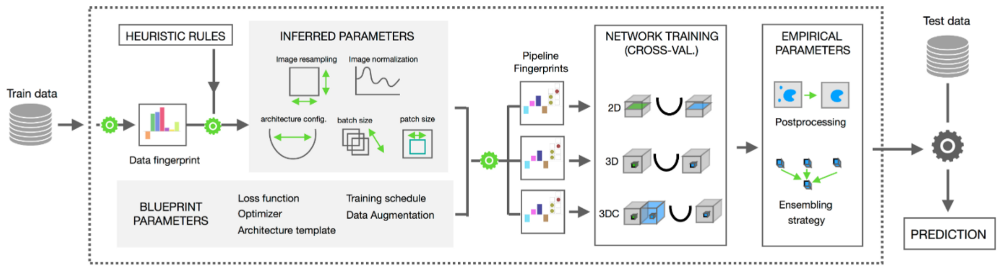


Figure 2: Dynamic U-Net (nnU-Net) end-to-end pipeline. The fingerprint of the input data is inferred through heuristic rules and is then merged with the inferred and blueprint parameters. The result is then passed to an ensemble of network configurations, which finds the best average Dice coefficient for the training set and selects the optimal configuration to use for predictions [22].

Hyperparameters : Dynamic U-Net was trained using the same parameters as U-Net, meaning it employed the Adam optimizer with a learning rate of 10^{-3}, a dropout of 0.5 and a batch size of 6. The Reduce Learning Rate on Plateau scheduler was also used on the initial learning rate. The loss was a 50%-50% weighted sum of the Dice loss and Cross entropy loss. Finally, Dynamic U-Net also had the maximum number of epochs set to 500 and used early stopping with a patience of 30 and a required improvement for each 30 epochs of at least 0.01.

2.3 Reproducible techniques

2.3.1 Lung segmentation

All the CTs from the 3 datasets were preprocessed using the R231CovidWeb model presented in [23]. This model is a U-Net(R231) that was specialized for COVID-19 CT scans and it is tasked with segmenting the lung from a chest CT scan. Therefore, given a full chest scan, it is able to automatically remove the tissue and bones adjacent to the lung and only keep the lungs themselves. This was to done to improve the lesion segmentation process and allow the models to focus strictly on the lungs. An additional benefit is the lower memory requirements of each preprocessed CT scan, compared to the original variant. An example of the preprocessing step can be seen in Figures 3,4, 5 and 6.

2.3.2 Data augmentation

Data augmentation was performed using sliding windows, noise filters and Maximum Intensity Projection (MIP). The sliding windows had a patch size of $192 \times 192 \times 16$ and an overlap of 0.5. They were applied to random fixed sized region crops ($256 \times 256 \times 16$) from the original CT, with the center being a foreground or background voxel based on the positive-negative ratio. As for the noise filters, they allow random Gaussian noise to be added with a probability of 15%. Finally, for MIP, every 3 adjacent slices in a CT are mapped into a single slice by selecting, for each pixel, the maximum intensity out of the 3 slices (see Figures 7, 8, 9 and 10). The number 3 in this case is known as the slab thickness of the MIP.

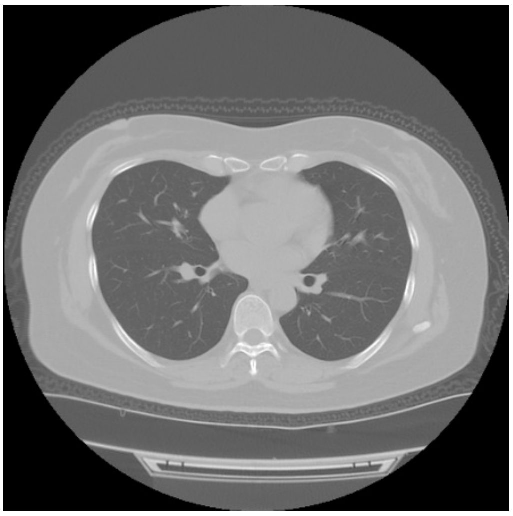


Figure 3: Example 1 of a thoracic CT scan seen from the transverse plane. This represents the output of a CT without any preprocessing.

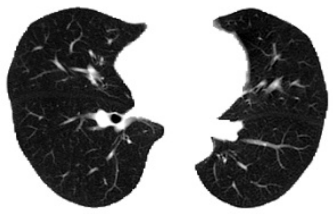


Figure 4: Preprocessed thoracic CT scan (Example 1). The lung is extracted from the rest of the tissue, reducing the size and complexity of the image.

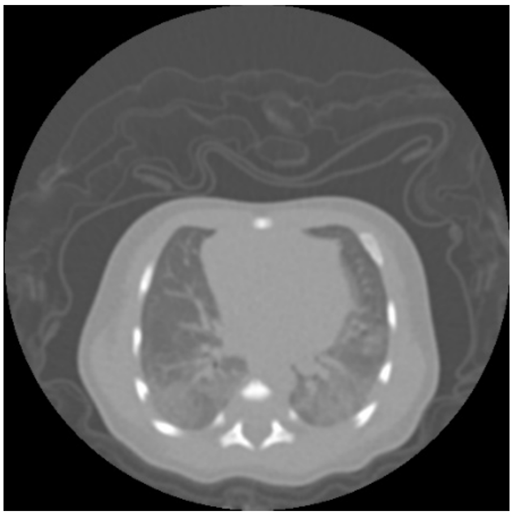


Figure 5: Example 2 of a thoracic CT scan seen from the transverse plane. This represents the output of a CT without any preprocessing. The scan itself has a lower quality than Example 1.

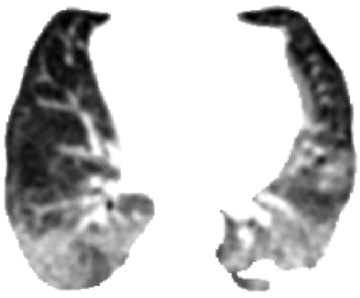


Figure 6: Preprocessed thoracic CT scan (Example 2). The lung is extracted from the rest of the tissue, reducing the size and complexity of the image. The lower quality is more visible.

2.3.3 Common evaluation metrics

In order to rate the quality of the segmentation, multiple distance based and threshold based evaluation metrics and techniques were employed. The main distance based metrics used are the volumetric and surface dice scores, as well as the Hausdorff distance (95% and 100%). These metrics were provided during the COVID-19-20 Grand Challenge and were adopted from [24]. As for the threshold based techniques, the model's performance will be shown via the AUC, sensitivity, specificity, precision and F1 scores. For each data augmentation technique, the computation time as well as the average GPU usage and full maximum memory usage were also calculated and compared.

2.4 Evaluation of the models

The two models were implemented using MONAI, which is an open source deep learning framework for healthcare imaging [1]. It is built on top of Pytorch, another open source framework specialized for machine learning [25]. To ensure reproducibility, the code of both models will be posted on the GitHub page of the University of Groningen. Calculation power is provided by the cluster computer of the University of Groningen, called Peregrine. Each model was trained using a single NVIDIA V100 Tensore Core GPU.

2.4.1 Training and model parameters

The main variables when it comes to training the two models are the usage of MIP techniques, the addition of noise filters and the inclusion of sliding windows. These variations were tested both separately and overlapping. To be more specific, the standard model variant used sliding windows as well as noise filters. All other model variants started from these specifications and differ in one or more aspects. For instance, the second variation does not use any sliding windows by setting both the crop size as well as the sliding window patch size to $192 \times 192 \times 16$. A third variation removes the noise entirely (i.e. 0% change of adding random Gaussian noise). Therefore, the two variants will be named No SW and No Noise from now on.

The following 3 variants use MIP preprocessing techniques applied to the CT scans. Therefore, the standard model was trained and tested on this MIP CTs. Furthermore, an additional variant tested whether removing the noise filters and applying MIP can improve the performance. Finally, a post-processing method was analyzed using the standard model trained with MIP CTs. This method overlapped the model's prediction with the ground truth and then used this overlapping prediction to analyze the performance. These 3 variations will be known as MIP, MIP No Noise and MIP Overlap.

In total, 6 model variants were tested, for both the U-Net and Dynamic U-Net models, as well as for both data split configurations. Other hyperparameters such as the learning rate, dropout or the batch size were optimized previous to the testing of the data augmentation techniques. This being said, the learning rate was set to 10^{-3}, the dropout to 0.5 and the batch size to 6. The Adam optimizer and the Reduce Learning Rate on Plateau scheduler were also used to further adapt the learning rate. As for the loss function, a weighted sum of the Dice loss and Cross entropy loss was employed, each accounting for 50% of the resulting loss. Finally, the models were allowed to train for a maximum of 500 epochs. However, early stopping with a patience of 30 and a required improvement for each 30 epochs of at least 0.01 prevented the model from training to this value.

2.4.2 Data Split

Two training and testing procedures were used for all model variants. Firstly, it is worth mentioning that the Grand Challenge data was separated into two datasets: a 200 CTs training set and a 50 CTs test set. Therefore, the first procedure involved training and validating the models using all the combined data apart from the Grand Challenge test set (279 CTs) and then testing the trained models on this test set (50 CTs). The second procedure involved taking 10% of the entire combined dataset as a holdout and using this as a test set. Thus, the rest of the 90% was used for training and validating. Selecting the

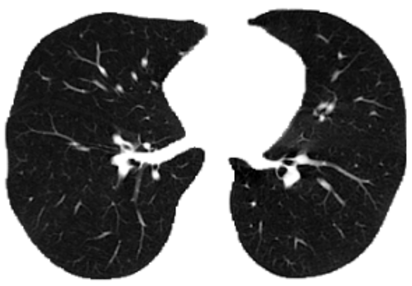


Figure 7: Preprocessed thoracic CT scan (Example 3) seen from the transverse plane.

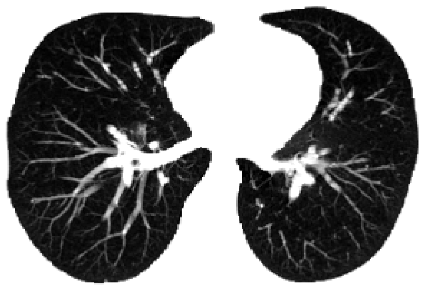


Figure 8: The thoracic CT scan from Example 3, preprocessed using MIP with a slab thickness of 3.

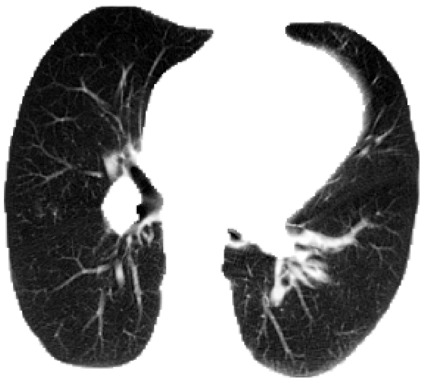


Figure 9: Preprocessed thoracic CT scan (Example 4) seen from the transverse plane.

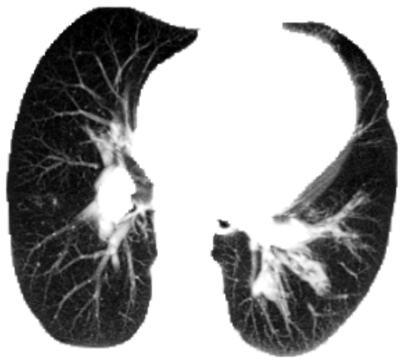


Figure 10: The preprocessed thoracic CT scan from Example 4 preprocessed using MIP with a slab thickness of 3.

holdout set was done randomly by picking 10% of data from each of the 3 initial datasets. For both of the procedures, 20% of the training dataset was used for validating the model's performance, leaving the rest of 80% for training itself.

3 Results

The results for all model variants can be seen in tables 3 and 4. These tables present all analyzed metrics for each model variation using the validation or the holdout datasets for testing. Therefore, they show the average volumetric dice (VD), average surface dice (SD), the Hausdorff distances (95% and 100%) as well as the AUC, sensitivity, specificity, precision and F1 scores.

It can be seen from the two tables that, for U-Net, not adding noise to the CTs seems to lead to the best results, although it is not a noticeable improvement from the standard model. The two variations have almost equal results for the holdout dataset (VD = 0.43, SD = 0.36, AUC = 0.77) while for the validation, not adding noise is a minor improvement (i.e. AUC = 0.8 rather than 0.79 for the standard model). Moreover, not using a sliding window or implementing the MIP data augmentation seems to slightly hurt the performance of U-Net. However, it is worth mentioning that the post-processing method of overlapping the model's prediction with the ground truth in the case of the MIP variant does seem to increase the volumetric and surface dice score, even though the AUC is smaller.

On the other hand, Dynamic U-Net seems to benefit from not having a sliding window, reaching the highest AUC of 0.81 out of all tested variants for the validation dataset. However, for the holdout dataset, not adding noise still leads to the best result with an AUC of 0.76. The standard model had a similar AUC of 0.75 but resulted in higher VD and SD (0.47 and 0.42 respectively), compared to the no noise variation (0.45 and 0.39 respectively). In line with the U-Net results, the post-processing overlap method does increase the VD and SD while still presenting a smaller AUC. All model variations, regardless of the dataset, have high specificity but the sensitivity and precision are quite low. This is also shown in the F1 score. The overlap post-processing method does improve the precision and F1 scores, but the sensitivity remains low.

Furthermore, tables 5 and 6 show the required training time as well as the average GPU usage and full maximum memory usage. Dynamic U-Net does seem to have on average a lower training time and a smaller full maximum memory usage, while retaining similar performance to U-Net. However, the GPU usage is noticeably higher for Dynamic U-Net.

Finally, Figures 11 to 18, present examples of lung CT scans along with the ground truth lesion segmentation and the corresponding predictions. More specifically, Figures 11, 13, 15, 17 show thoracic preprocessed CT scans where the lung was extracted from the rest of the CT. The green color corresponds to the lesion ground truth while the red and blue colors to the prediction of the U-Net (without noise) and Dynamic U-Net (without a sliding window), respectively. Similarly, Figures 12, 14, 16, 18 contain the extracted lung on which MIP was applied. The green section represents the ground truth of the lesion, while the red and blue annotations correspond to the prediction of the U-Net (without noise) and Dynamic U-Net (without a sliding window) models, respectively. These models were thus trained using MIP CT scans.

U-Net	VD	SD	Haus (100%)	Haus (95%)	AUC	Sens	Spec	Prec	F1
Standard	0.53	0.43	130.96	74.26	0.79	0.6	0.99	0.44	0.51
No SW	0.49	0.38	132.24	76.93	0.74	0.49	0.99	0.47	0.48
No Noise	**0.53**	**0.43**	**132.98**	**75.49**	**0.80**	**0.60**	**0.99**	**0.44**	**0.51**
MIP	0.48	0.36	148.08	85.95	0.75	0.51	0.99	0.32	0.44
MIP No Noise	0.48	0.34	152.8	85.31	0.74	0.50	0.99	0.39	0.44
MIP Overlap	0.66	0.55	198.73	110.39	0.75	0.51	1.0	1.0	0.67
Dynamic U-Net	VD	SD	Haus (100%)	Haus (95%)	AUC	Sens	Spec	Prec	F1
Standard	0.54	0.46	135.21	71	0.74	0.48	0.99	0.62	0.54
No SW	**0.59**	**0.53**	**140.24**	**69.93**	**0.81**	**0.63**	**0.99**	**0.56**	**0.59**
No Noise	0.46	0.37	225.74	97.37	0.72	0.45	0.99	0.49	0.47
MIP	0.54	0.43	139.81	69.42	0.76	0.54	0.99	0.54	0.54
MIP No Noise	0.51	0.4	145.48	71.75	0.76	0.52	0.99	0.5	0.51
MIP Overlap	0.67	0.57	174.27	83.14	0.77	0.54	1.0	1.0	0.7

Table 3: The results of U-Net and Dynamic U-Net when labeling CT scans from the validation dataset. For U-Net, not adding any noise and using sliding windows lead to the highest AUC of 0.80. Dynamic U-Net benefited from noise and no sliding windows, reaching the AUC of 0.81.

U-Net	VD	SD	Haus (100%)	Haus (95%)	AUC	Sens	Spec	Prec	F1
Standard	**0.43**	**0.36**	**171.46**	**120.05**	**0.77**	**0.55**	**0.99**	**0.33**	**0.41**
No SW	0.41	0.35	169.55	109.23	0.72	0.46	0.99	0.37	0.41
No Noise	**0.43**	**0.36**	**172.20**	**120.17**	**0.77**	**0.55**	**0.99**	**0.33**	**0.41**
MIP	0.38	0.30	161.54	113.52	0.73	0.47	0.99	0.30	0.37
MIP No Noise	0.38	0.30	160.62	112.54	0.73	0.46	0.99	0.31	0.37
MIP Overlap	0.61	0.55	122.13	48.25	0.73	0.47	1.0	1.0	0.63
Dynamic U-Net	VD	SD	Haus (100%)	Haus (95%)	AUC	Sens	Spec	Prec	F1
Standard	**0.47**	**0.42**	**171.08**	**112.32**	**0.75**	**0.50**	**0.99**	**0.42**	**0.46**
No SW	0.41	0.36	172.49	123.51	0.73	0.46	0.99	0.36	0.41
No Noise	**0.45**	**0.39**	**230.79**	**136.84**	**0.76**	**0.54**	**0.99**	**0.37**	**0.44**
MIP	0.38	0.31	164.75	100.85	0.69	0.39	0.99	0.37	0.38
MIP No Noise	0.34	0.29	212.29	104.23	0.67	0.35	0.99	0.35	0.35
MIP Overlap	0.56	0.48	116.87	43.24	0.69	0.39	1.0	1.0	0.56

Table 4: The results of U-Net and Dynamic U-Net when labeling CT scans from the holdout dataset. For U-Net, using sliding windows, with or without noise, leads to the highest AUC of 0.77. The same variations also make Dynamic U-Net reach the highest AUC of 0.76 or 0.75.

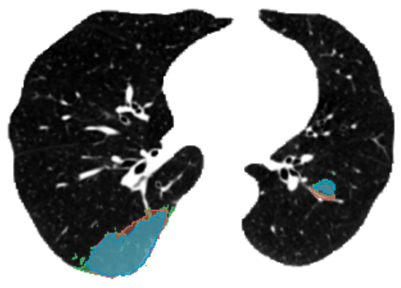


Figure 11: Preprocessed thoracic CT scan 1 along with the ground truth (green), U-Net No Noise Prediction (red) and Dynamic U-Net No SW prediction (blue)

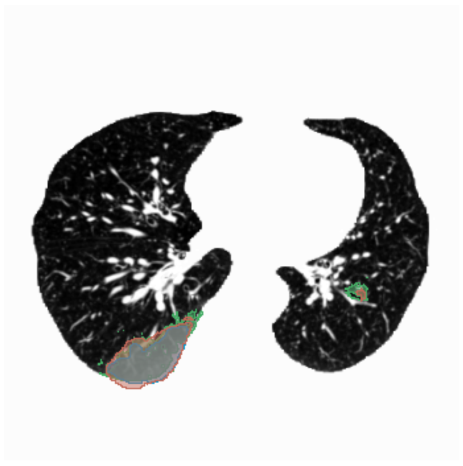


Figure 12: MIP CT scan 1 (slab thickness 3) along with the ground truth (green), U-Net No Noise Prediction (red) and Dynamic U-Net No Noise prediction (blue)

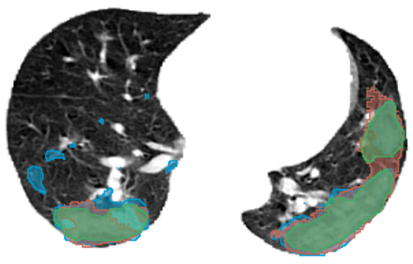


Figure 13: Preprocessed thoracic CT scan 2 along with the ground truth (green), U-Net No Noise Prediction (red) and Dynamic U-Net No SW prediction (blue)

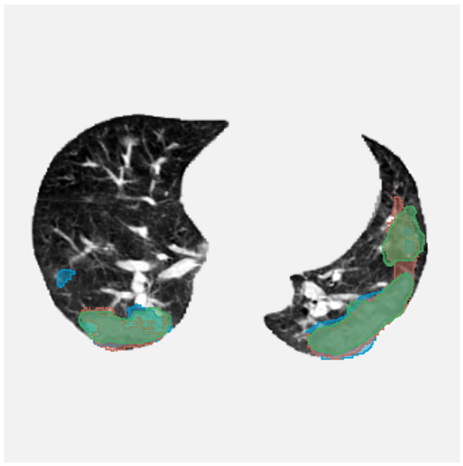


Figure 14: MIP CT scan 2 (slab thickness 3) along with the ground truth (green), U-Net No Noise Prediction (red) and Dynamic U-Net No Noise prediction (blue)

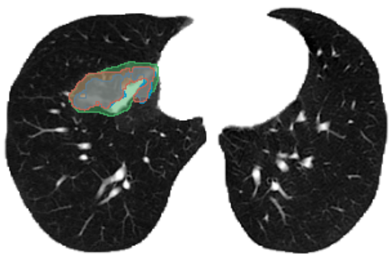


Figure 15: Preprocessed thoracic CT scan 3 along with the ground truth (green), U-Net No Noise Prediction (red) and Dynamic U-Net No SW prediction (blue)

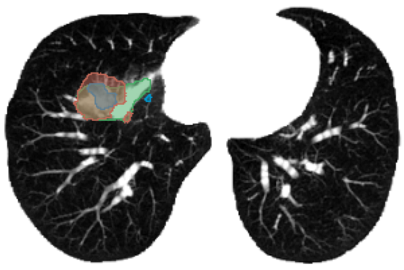


Figure 16: MIP CT scan 3 (slab thickness 3) along with the ground truth (green), U-Net No Noise Prediction (red) and Dynamic U-Net No Noise prediction (blue)

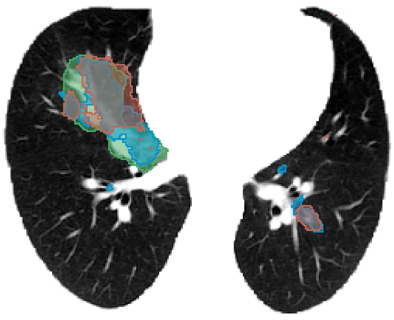


Figure 17: Preprocessed thoracic CT scan 4 along with the ground truth (green), U-Net No Noise Prediction (red) and Dynamic U-Net No SW prediction (blue)

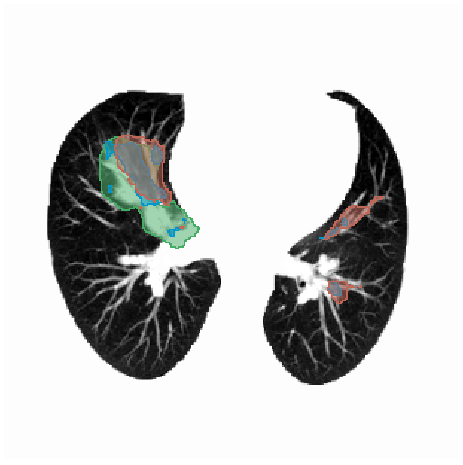


Figure 18: MIP CT scan 4 (slab thickness 3) along with the ground truth (green), U-Net No Noise Prediction (red) and Dynamic U-Net No Noise prediction (blue)

Training Time & GPU Usage	U-Net			Dynamic U-Net		
	Time	Util*	Mem**	Time	Util*	Mem**
Standard	16h 30m	43.7%	41.33G	16h 5m	57.1%	38.46G
No SW	18h 1m	35.1%	41.40G	16h 37m	No metrics	38.54G
No Noise	17h 59m	43.1%	41.82G	9h 5m	77.9%	38.49G
MIP	13h 58m	46.3%	40.26G	15h 31m	No metrics	37.24G
MIP No Noise	20h 52m	30.0%	39.89G	17h 15m	56.9%	36.98G

Table 5: U-Net and Dynamic U-Net training times and GPU usage for the validation dataset. * average GPU usage, ** Full Max Memory Usage

Training Time & GPU Usage	U-Net			Dynamic U-Net		
	Time	Util*	Mem**	Time	Util	Mem
Standard	16h 8m	44.7%	43.09G	10h 36m	83.5%	40.18G
No SW	13h 36m	33.8%	43.43G	9h 53m	50.1%	39.82G
No Noise	16h 6m	No metrics	43.52G	10h 6m	80.9%	40.35G
MIP	14h 9m	43.0%	41.57G	8h 21m	81.1%	39.0G
MIP No Noise	11h 13m	No metrics%	41.68G	10h 13m	83.6%	39.89G

Table 6: U-Net and Dynamic U-Net training times and GPU usage for the holdout dataset. * average GPU usage, ** Full Max Memory Usage

4 Discussion

All in all, this thesis analyzed the effect of three data augmentation techniques (i.e. sliding windows, noise filters, MIP) on the performance of two CNN models (U-Net and Dynamic U-Net). This involved using reproducible frameworks with publicly available model architectures and data. Analyzing the results, it can be seen that the standard model which uses a sliding window and does not apply noise to the CT scans seems to show the most promising results for the used dataset. This is the case for both U-Net and Dynamic U-Net. However, noise filters are a common data augmentation technique which often increases neural network models' generalization capabilities and reduce overfitting [26]. For this reason, it might be advisable to still include such filters in future research, especially since the performance difference between the standard and the no noise variations was not that significant. As for the maximum intensity projection techniques, they did not seem to be beneficial in segmenting the COVID-19 lesions, neither for U-Net nor the Dynamic U-Net. In both cases, the performance was slightly hurt by the addition of this technique. However, it might be possible that different specifications for the MIP method would lead to better results. For instance, the current analysis implemented MIP by using 3 adjacent CT scan slices. It might be the case that increasing or decreasing this number would solve the shortcoming of the method.

As for the difference between U-Net and Dynamic U-Net, the performance results are quite similar. Some variations seem to favor one of the two architectures, but the differences are not that noticeable. Dynamic U-Net does seem to reach the highest AUC for the validation dataset when not including a sliding window. On the other hand, U-Net has the highest AUC for the holdout dataset, either for the standard or the no noise variations. Overall, the performance difference between each architecture's most optimal variation is not significant. However, Dynamic U-Net seems to use less memory and require less training time. This should be put in the context of early stopping, which stopped training after 30 epochs of no improvements. Therefore, for the weaker variations, the model stopped training sooner because no improvement was found. Despite this, Dynamic U-Net required less training time, even for the variations that surpassed U-Net's results. This might show that Dynamic U-Net is more versatile and can adapt easier to new problems.

When analyzing the predictions of both architectures, some aspects stood out. Firstly, there were cases in which the models overestimated the lesion, even surpassing the boundaries of the lungs. This might suggest that the models are not fully adapted to the required lesion segmentation task. On the other hand, other predictions highly underestimated the lesions of the more severe cases of COVID infection. This might be in part explained by the fact that some cases also presented comorbidities, such as interstitial lung disease or fibrosis.

A final aspect which should be discussed is the quality of the used data. On closer inspection, it was observed that both the COVID-19-20 Grand Challenge data and the Moscow data had a low image quality with respect to the longitudinal plane. This can be observed in Figures 19 and 20, when comparing them with a longitudinal CT scan from the Coronacases dataset in Figure 21, which is sharper and has more details. In addition, the Grand Challenge dataset also presents lower quality CT scans seen from the transverse plane. This can be seen in Figure 22, whose quality is visibly lower than Figures 7 and 9 which were selected from the other datasets. Furthermore, it was observed that some labels from the Grand Challenge dataset seemed to have missed several visible lesions seen on the original CT. It might be the case that those lesions are not associated to a COVID-19 infection, however, more analyses should be done in this regard. These aspects should be taken into consideration when interpreting the results, since the majority of the used data came from the two datasets.

4.1 Limitations

When considering these results, it must be stressed that the amount of training and testing data available was not high. It might be the case that one of the presented techniques requires more data in order to truly show its potential. Therefore, the current analysis should be repeated, when more data is gathered. In addition, the new data should be more diverse by being collected from multiple hospitals and countries. This is one of the limitations of the presented analysis, since most of the already limited data was gathered from only 2 sources. Therefore, even if the models had achieved great performance, they might have been unreplicable on new data from different regions. The main reason is that different medical facilities use a diverse range of CT scanners, which may produce slightly inconsistent data. Furthermore, the quality of the CT scans and the labels might prove to be an additional limitation which prevented the models from reaching their true potential.

5 Conclusion and future work

Although maximum intensity projection did not lead to improvements in the performance of the models, this is still a technique that should be further researched. It might be the case that it would prove more useful in classification tasks which were not covered in this analysis. Moreover, further specifications of

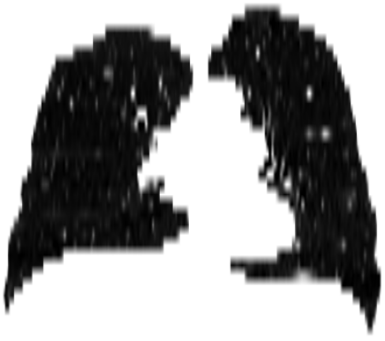


Figure 19: Low quality preprocessed thoracic CT scan seen from the longitudinal plane, belonging to the Moscow dataset

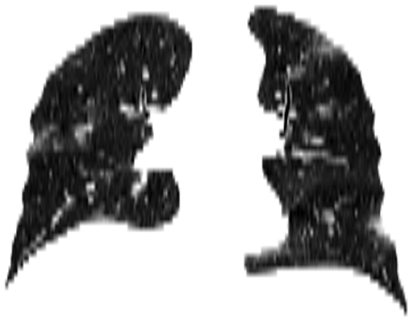


Figure 20: Low quality preprocessed thoracic CT scan seen from the longitudinal plane, belonging to the Grand Challenge dataset

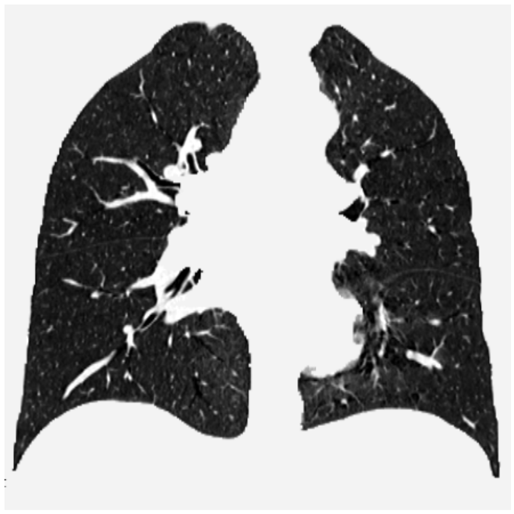


Figure 21: High quality preprocessed thoracic CT scan seen from the transverse plane, belonging to the Coronacases dataset

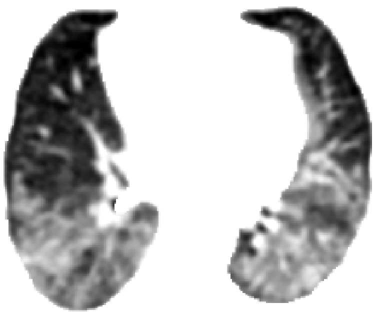


Figure 22: Low quality preprocessed thoracic CT scan seen from the transverse plane, belonging to the Grand Challenge dataset

MIP should be tested in order to analyze whether increasing or decreasing the number of adjacent slices that are merged improves the efficiency of the models. In addition, maximum intensity projection should be compared to its counterpart, minimum intensity projection, to see whether one has the advantage over the other.

Considering one of the goal of this thesis was to use reproducible pathways such as open-source data, future research should aim to increase the amount of publicly available, high-quality datasets. This could be done with the help of professional radiologists, which were not accommodated in the current thesis due to the high demand in radiologists throughout the pandemic. However, the aim is to include them in future studies. An additional goal for further work would be to employ the open-source image labeling and learning tool, MONAI Label [27], in the creation of new open-source datasets. This tool allows professional radiologists to create AI annotation models that learn the annotation task directly from the radiologists' input. The models are then able to automatically create new annotations according to the user's preferences. To be more specific, the tool is automatically and continuously trained on manually labeled data created by the radiologist. Once the performance becomes satisfactory, the user can let the model create annotations for the new data and is able to manually adjust the automatic annotations in case any mislabeling was made. These manual adjustments are then further learned by the model. The tool should therefore greatly reduce the effort and time radiologists put into annotating data, thus leading to an increase in productivity and availability of new data. The interface of this tool can be seen in Figure 23. Additionally, the thesis also provides a setup guide for the MONAI Label tool in the appendix (see Section A).

All in all, the data augmentation techniques presented in this thesis should be further researched, both for segmentation and classification tasks. The reason for this is that they present an intelligent alternative to simply increasing the computational power of neural networks. Ideally, such analyses would use larger and higher quality open-source datasets than what was available in this thesis. Furthermore, a serious effort should be put into creating new high-quality, open-source datasets. This would allow more researchers to contribute to the advancements in medical imaging by using reproducible pathways that can be efficiently peer-reviewed.

6 Ethics

All data used in this thesis is completely public and anonymized. There is no conflict of interest to disclose. However, the usage of neural networks in medical imaging does come with some ethical considerations. Firstly, neural network models lack explainability and interpretability because of their black box nature. Therefore, physicians cannot verify what features led to the model's decision, either for segmentation or classification. This is especially troublesome in the medical sector since it involves high stake decisions about people's lives. Secondly, in cases where the physicians and the model contradict each other, an ethical discussion arises for who should be responsible in case of a misdiagnosis, the physician, the hospital or the model's developers? Finally, machine learning models often present biases because of the data and settings they were trained with. In this way, certain minorities might be poorly represented, which would decrease the model's reliability in their case. These ethical aspects should be taken into consideration when introducing machine learning models in the med-

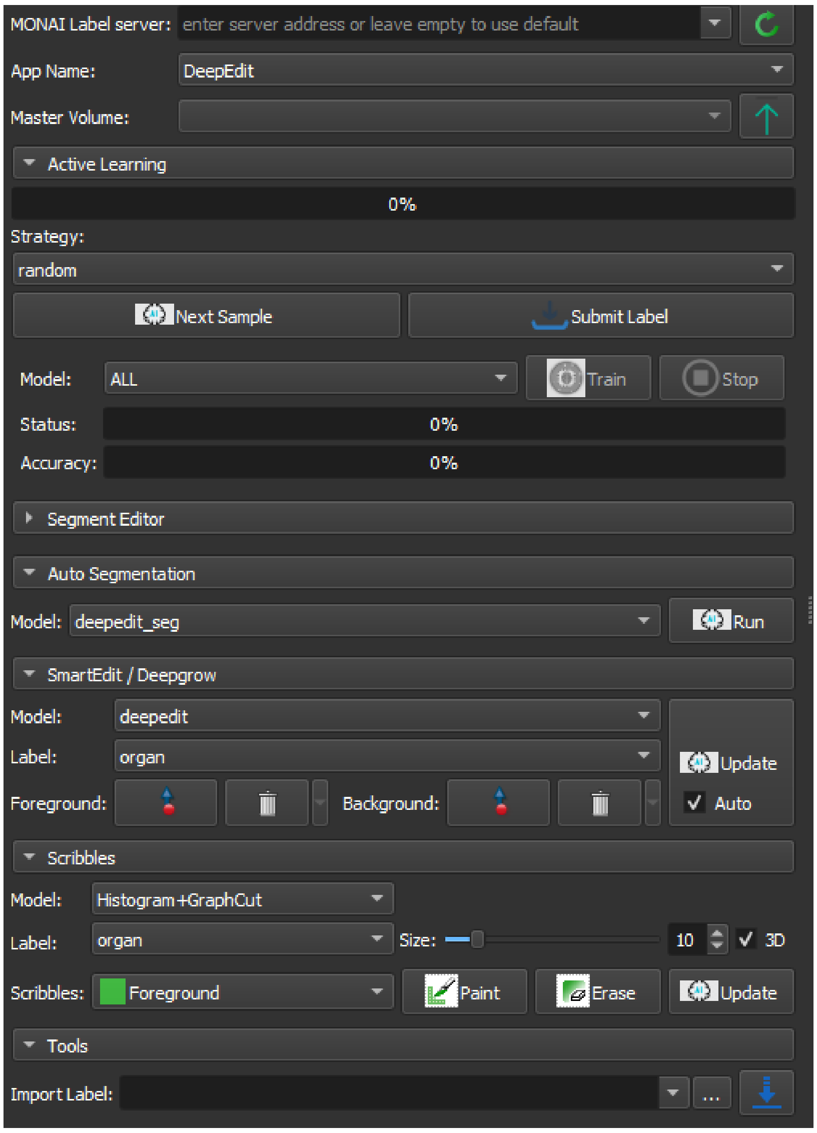

MONAI Label server: enter server address or leave empty to use default
App Name: DeepEdit
Master Volume:
Strategy: random
Model: ALL
Status: 0%
Accuracy: 0%
Model: deepedit_seg
Model: deepedit
Label: organ
✓ Auto
Model: Histogram+GraphCut
Label: organ
Size: 10 ✓ 3D
Scribbles: 🟩 Foreground
Import Label:

Figure 23: The working interface of the MONAI Label Tool [27]

ical sector to ensure the fairness and reliability of the medical diagnostics.

7 Acknowledgement

We thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high performance computing cluster.

8 Additional Contributions

As part of this thesis, the poster seen in Appendix B was also presented during the Honours College Symposium.

References

[1] T. M. Consortium, "Project monai," Dec. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4323059

[2] F. Shi, J. Wang, J. Shi, Z. Wu, Q. Wang, Z. Tang, K. He, Y. Shi, and D. Shen, "Review of artificial intelligence techniques in imaging data acquisition, segmentation, and diagnosis for covid-19," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 4–15, 2021.

[3] A. Borakati, A. Perera, J. Johnson, and T. Sood, "Diagnostic accuracy of x-ray versus ct in covid-19: a propensity-matched database study," *BMJ Open*, vol. 10, no. 11, 2020. [Online]. Available: https://bmjopen.bmj.com/content/10/11/e042946

[4] E. Benmalek, J. Elmhamdi, and A. Jilbab, "Comparing ct scan and chest x-ray imaging for covid-19 diagnosis," *Biomedical Engineering Advances*, vol. 1, p. 100003, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667099221000037

[5] ——, "Comparing ct scan and chest x-ray imaging for covid-19 diagnosis," *Biomedical Engineering Advances*, vol. 1, p. 100003, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667099221000037

[6] M. Karam, S. Althuwaikh, M. Alazemi, A. Abul, A. Hayre, A. Alsaif, and G. Barlow, "Chest ct versus rt-pcr for the detection of covid-19: systematic review and meta-analysis of comparative studies," *JRSM Open*, vol. 12, no. 5, p. 20542704211011837, 2021, pMID: 34035931. [Online]. Available: https://doi.org/10.1177/20542704211011837

[7] V. Mehta, D. Jyoti, R. T. Guria, and C. B. Sharma, "Correlation between chest ct and rt-pcr testing in india's second covid-19 wave: a retrospective cohort study," *BMJ evidence-based medicine*, January 2022.

[8] M. Savic, Y. Ma, G. Ramponi, W. Du, and Y. Peng, "Lung nodule segmentation with a region-based fast marching method," *Sensors*, vol. 21, no. 5, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/5/1908

[9] R. Robinson, Q. Dou, D. Coelho de Castro, K. Kamnitsas, M. de Groot, R. M. Summers, D. Rueckert, and B. Glocker, "Image-level harmonization of multi-site data using image-and-spatial transformer networks," *Lecture Notes in Computer Science*, p. 710–719, 2020. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-59728-3_69

[10] H. Hassan, Z. Ren, H. Zhao, S. Huang, D. Li, S. Xiang, Y. Kang, S. Chen, and B. Huang, "Review and classification of ai-enabled covid-19 ct imaging models based on computer vision tasks," *Computers in Biology and Medicine*, vol. 141, p. 105123, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482521009173

[11] G. Wang, X. Liu, C. Li, Z. Xu, J. Ruan, H. Zhu, T. Meng, K. Li, N. Huang, S. Zhang, and et al., "A noise-robust framework for automatic segmentation of covid-19 pneumonia lesions from ct images," *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, p. 2653–2663, 2020.

[12] I. Shiri, H. Arabi, Y. Salimi, A. H. Sanaat, A. Akhavanalaf, G. Hajianfar, D. Askari, S. Moradi, Z. Mansouri, M. Pakbin, S. Sandoughdaran, H. Abdollahi, A. R. Radmard, K. Rezaei-Kalantari, M. G. Oghli, and H. Zaidi, "Coli-net: Fully automated covid-19 lung and infection pneumonia lesion detection and segmentation from chest ct images," *medRxiv*, 2021. [Online]. Available: https://www.medrxiv.org/content/early/2021/04/13/2021.04.08.21255163

[13] X. Xu, Y. Wen, L. Zhao, Y. Zhang, Y. Zhao, Z. Tang, Z. Yang, and C. Y.-C. Chen, "Cares-unet: Content-aware residual unet for lesion segmentation of covid-19 from chest ct images," *Medical Physics*, vol. 48, no. 11, pp. 7127–7140, 2021. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/mp.15231

[14] X. Wang, L. Jiang, L. Li, M. Xu, X. Deng, L. Dai, X. Xu, T. Li, Y. Guo, Z. Wang, and P. L. Dragotti, "Joint learning of 3d lesion segmentation and classification for explainable covid-19 diagnosis," *IEEE TRANSACTIONS ON MEDICAL IMAGING*, vol. 40, no. 9, pp. 2463–2476, SEP 2021.

[15] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, and et al., "The cancer imaging archive (tcia): Maintaining and operating a public information repository," *Journal of Digital Imaging*, vol. 26, no. 6, p. 1045–1057, 2013.

[16] H. Roth, Z. Xu, C. T. Diez, R. S. Jacob, J. Zember, J. Molto, W. Li, S. Xu, B. Turkbey, E. Turkbey, and et al., "Rapid artificial intelligence solutions in a pandemic - the covid-19-20 lung ct lesion segmentation challenge," 2021.

[17] "Ct images in covid-19." [Online]. Available: https://doi.org/10.7937/TCIA.2020.GQRY-NC81

[18] S. P. Morozov, A. E. Andreychenko, N. A. Pavlov, A. V. Vladzymyrskyy, N. V. Ledikhova, V. A. Gombolevskiy, I. A. Blokhin, P. B. Gelezhe, A. V. Gonchar, and V. Y. Chernina, "Mosmeddata: Chest ct scans with covid-19 related findings dataset," 2020. [Online]. Available: https://arxiv.org/abs/2005.06465

[19] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Q. Duong, and M. Ghassemi, "Covid-19 image data collection: Prospective predictions are the future," *arXiv 2006.11988*, 2020. [Online]. Available: https://github.com/ieee8023/covid-chestxray-dataset

[20] M. Jun, G. Cheng, W. Yixin, A. Xingle, G. Jiantao, Y. Ziqi, Z. Minqing, L. Xin, D. Xueyuan, C. Shucheng, W. Hao, M. Sen, Y. Xiaoyu, N. Ziwei, L. Chen, T. Lu, Z. Yuntao, Z. Qiongjie, D. Guoqiang, and H. Jian, "Covid-19 ct lung and infection segmentation dataset," Apr. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3757476

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[22] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-net: a self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, dec 2020. [Online]. Available: https://doi.org/10.1038%2Fs41592-020-01008-z

[23] J. Hofmanninger, F. Prayer, J. Pan, S. Röhrich, H. Prosch, and G. Langs, “Automatic lung segmentation in routine imaging is primarily a data diversity problem, not a methodology problem,” *European Radiology Experimental*, vol. 4, no. 1, Aug 2020. [Online]. Available: http://dx.doi.org/10.1186/s41747-020-00173-2

[24] M. Antonelli, A. Reinke, S. Bakas, K. Farahani, AnnetteKopp-Schneider, B. A. Landman, G. Litjens, B. Menze, O. Ronneberger, R. M. Summers, B. van Ginneken, M. Bilello, P. Bilic, P. F. Christ, R. K. G. Do, M. J. Gollub, S. H. Heckers, H. Huisman, W. R. Jarnagin, M. K. McHugo, S. Napel, J. S. G. Pernicka, K. Rhode, C. Tobon-Gomez, E. Vorontsov, H. Huisman, J. A. Meakin, S. Ourselin, M. Wiesenfarth, P. Arbelaez, B. Bae, S. Chen, L. Daza, J. Feng, B. He, F. Isensee, Y. Ji, F. Jia, N. Kim, I. Kim, D. Merhof, A. Pai, B. Park, M. Perslev, R. Rezaiifar, O. Rippel, I. Sarasua, W. Shen, J. Son, C. Wachinger, L. Wang, Y. Wang, Y. Xia, D. Xu, Z. Xu, Y. Zheng, A. L. Simpson, L. Maier-Hein, and M. J. Cardoso, “The medical segmentation decathlon,” 2021. [Online]. Available: https://arxiv.org/abs/2106.05735

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[26] H. Noh, T. You, J. Mun, and B. Han, “Regularizing deep neural networks by noise: Its interpretation and optimization,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[27] A. Diaz-Pinto, S. Alle, A. Ihsani, M. Asad, V. Nath, F. Pérez-García, P. Mehta, W. Li, H. R. Roth, T. Vercauteren, D. Xu, P. Dogra, S. Ourselin, A. Feng, and M. J. Cardoso, “Monai label: A framework for ai-assisted interactive labeling of 3d medical images,” 2022. [Online]. Available: https://arxiv.org/abs/2203.12362

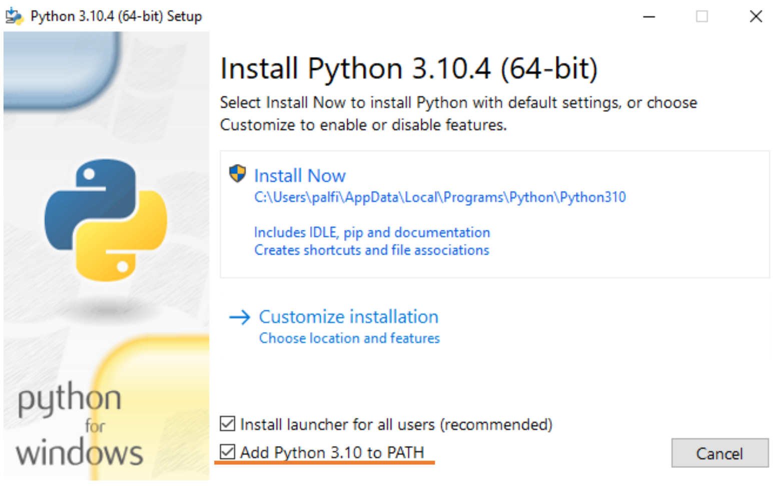
A Appendix

MONAI Label Setup

Bogdan Palfi (palfibogdan2@gmail.com)

1. Download the latest version of Python: https://www.python.org/downloads/

2. Install Python by opening the downloaded file
   a) Make sure the “Add Python *version number* to PATH” is checked
   b) Click on Install Now



Python 3.10.4 (64-bit) Setup

Install Python 3.10.4 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\palfi\AppData\Local\Programs\Python\Python310

Includes IDLE, pip and documentation
Creates shortcuts and file associations

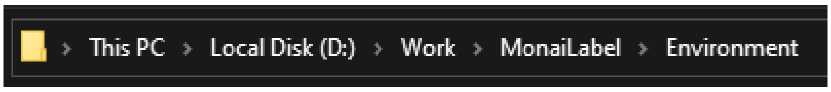
Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.10 to PATH

3. Create a virtual environment in Python so that every Python package is in the same place: https://docs.python.org/3/library/venv.html
   a) Go into Window`s “Type here to search” bar
   b) Search for “Command Prompt” terminal
   c) In the opened terminal type:

```
python3 -m venv /path/to/new/virtual/environment
```

   Note 1: if python3 is not recognized, just write python instead (without 3), along with the rest of the text
   Note 2: “/path/to/new/virtual/environment” represents the folder location in which you wish to install the environment (for instance D:\Work\MonaiLabel\Environment)



4. With the virtual environment installed, in the terminal, navigate to the “Scripts” folder inside the environment folder and activate the environment
   Note: this step should always be done before starting the MONAI Label server
   a) In the terminal, type either **D:** or **C:** (i.e. the name of the disk in which the environment folder is: disk D or disk C)

b) To move to the Scripts folder type in the terminal:

   cd /path/to/new/virtual/environment/Scripts

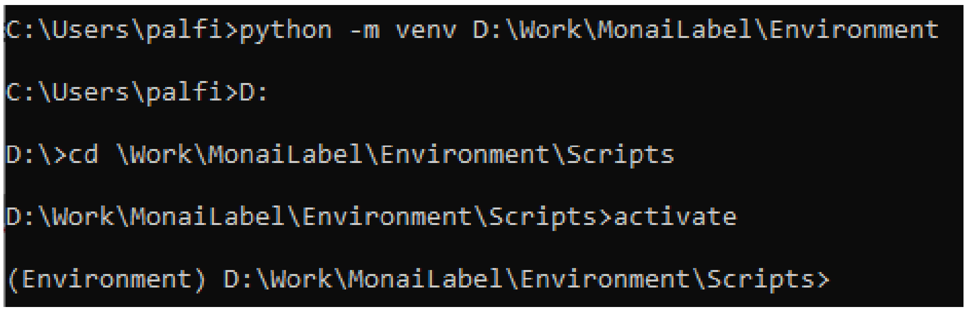
   For instance: cd \Work\MonaiLabel\Environment\Scripts

   Note 1: in terminals, "cd" is used to specify the folder in which you want to go to.

   cd \desired_folder

   Note 2: in terminals, "cd.." is used to go from the current folder to the parent folder (i.e. one above in the hierarchy)

c) When inside the Scripts folder, type the word: activate
   The name of the Environment should appear in front of the path: (Environment)



```
C:\Users\palfi>python -m venv D:\Work\MonaiLabel\Environment

C:\Users\palfi>D:

D:\>cd \Work\MonaiLabel\Environment\Scripts

D:\Work\MonaiLabel\Environment\Scripts>activate

(Environment) D:\Work\MonaiLabel\Environment\Scripts>
```

5. Install MONAI Label, download the wanted app and activate the server:
   https://docs.monai.io/projects/label/en/latest/quickstart.html
   Note: make sure the environment (step 4) is already activated

   a) Type in the Command Prompt terminal:

      pip install monailabel

   b) Once MONAI Label is installed, create a new folder for the project itself (for instance, in the MonaiLabel folder, create the folder Project. Now, you would have two folder in the MonaiLabel folder, the Project and the Environment folders)
   c) Navigate to this new folder inside the terminal. Type:
      cd..
      cd..
      cd Project

   d) Download the MONAI Label app you wish to use (e.g. deepedit, Radiology). Type:
      monailabel apps --name deepedit --download --output .

      Note: after the parameter --name, you can write either "deepedit" or "radiology", depending on which app you wish to use. Make sure to include the dot at the end, which specifies to download the app in the current folder

e) Inside the Project folder, create a new folder “dataset”. Either through File Explorer from Windows (Right Click – New Folder – and give it the name “dataset”). Or type in the terminal: mkdir dataset

f) Place your CTs inside the dataset folder

g) Start the MONAI Label server. Type:

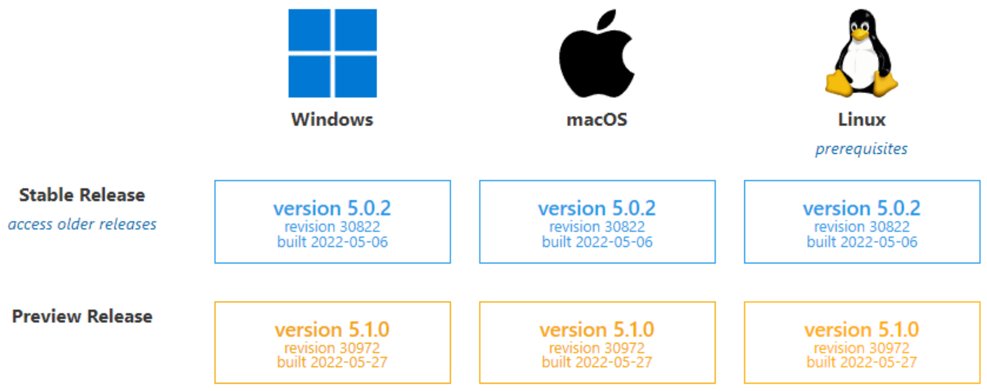
```
monailabel start_server --app deepedit --studies dataset --conf models deepedit
```

Note 1: if using radiology, replace deepedit with radiology in the command above

Note 2: for the --studies parameter, you need to specify the path to the dataset folder in which you have the CTs (in this case, we only write dataset since it’s the name of the dataset folder)

6. Open 3D Slicer, download the MONAI Label extension, add the data and start annotating

Note: make sure the version of 3D Slicer is at least 5.1 (so choose the Preview Release)



	Windows	macOS	Linux *prerequisites*
Stable Release *access older releases*	version 5.0.2 revision 30822 built 2022-05-06	version 5.0.2 revision 30822 built 2022-05-06	version 5.0.2 revision 30822 built 2022-05-06
Preview Release	version 5.1.0 revision 30972 built 2022-05-27	version 5.1.0 revision 30972 built 2022-05-27	version 5.1.0 revision 30972 built 2022-05-27

a) Open 3D Slicer

b) Install MONAI Label Extension:
View -> **Extensions Manager,** and search for MONAI Label
You might need to restart the 3D Slicer App. Once restarted,

You should have the logo in the toolbar.

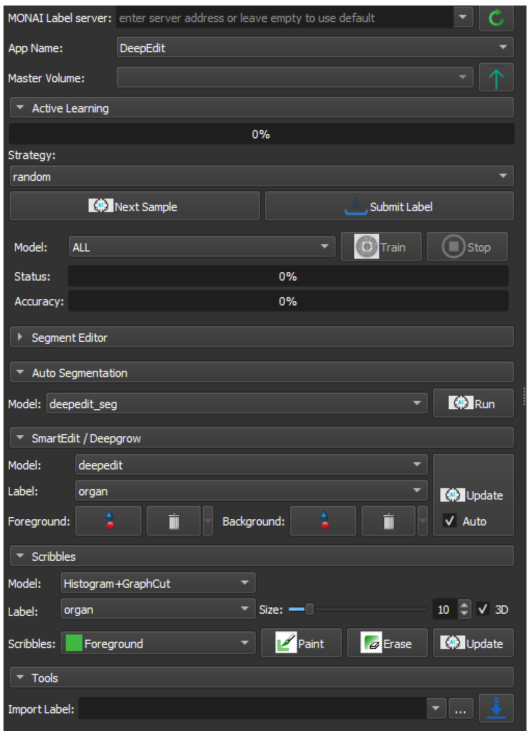
MONAILabel
Active Learning
This is Active Learning solution developed under project MONAILabel

c) Add the CTs: **File** -> Add Data

Or simply click the Add Data button

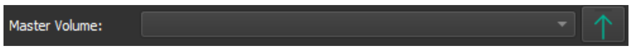
d) Press the button and then the refresh button (**Note: the MONAI Label server must be running for this refresh button to work**)

If everything is working, you should see this interface->



MONAI Label server: enter server address or leave empty to use default
App Name: DeepEdit
Master Volume:
Active Learning
0%
Strategy:
random
Next Sample
Submit Label
Model: ALL
Train
Stop
Status: 0%
Accuracy: 0%
Segment Editor
Auto Segmentation
Model: deepedit_seg
Run
SmartEdit / Deepgrow
Model: deepedit
Label: organ
Foreground:
Background:
Update
✓ Auto
Scribbles
Model: Histogram+GraphCut
Label: organ
Size: 10
✓ 3D
Scribbles: Foreground
Paint
Erase
Update
Tools
Import Label:
...

e) Click the Upload Volume button:



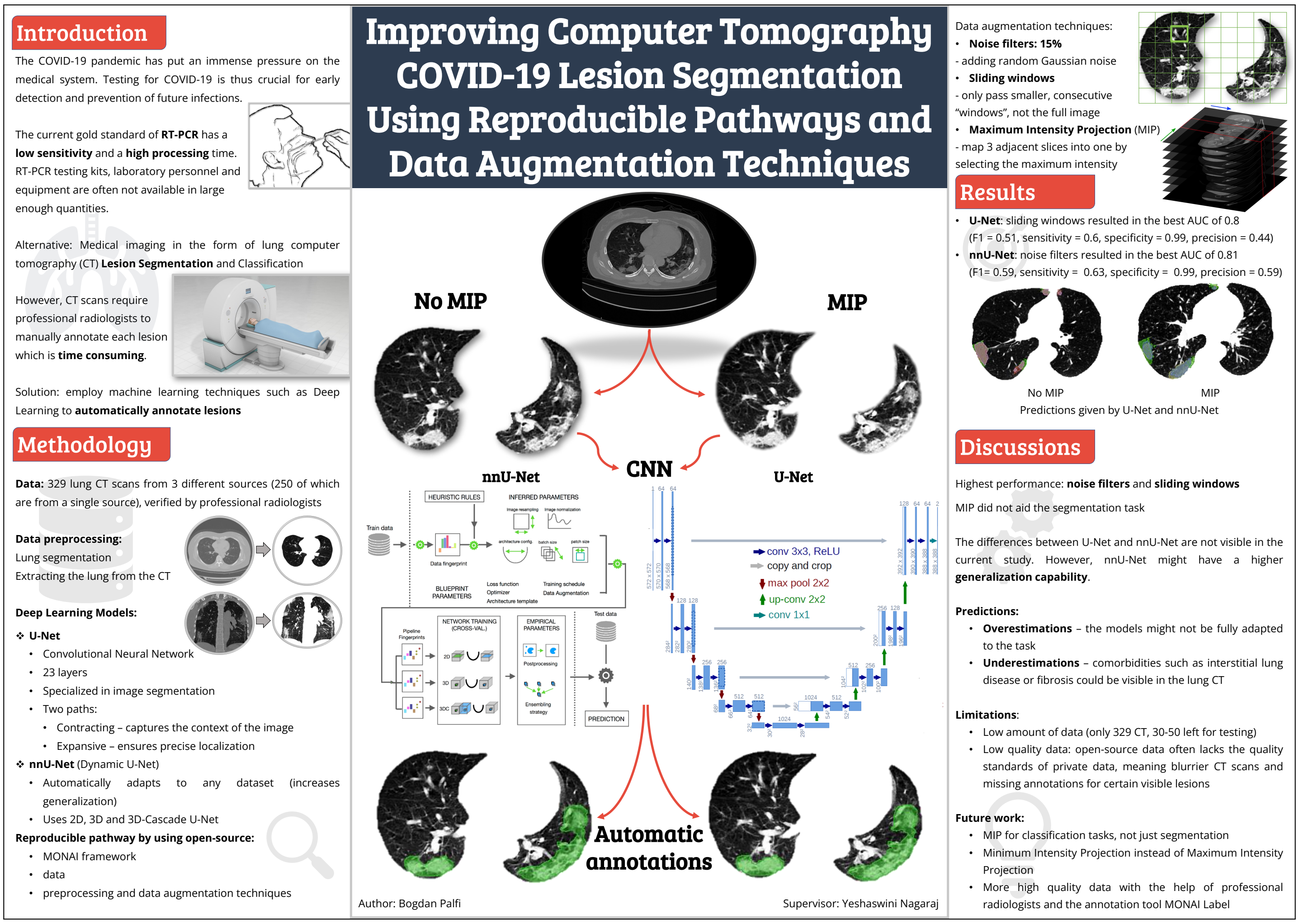
f) You should now be able to either create the annotations using the Scribble functions or to upload an annotation using the Tools -> Import Label

g) Once you are satisfied with the annotation, click on: Submit Label and then move to the Next Sample

h) Press **Train** to train the model on the CTs and labels you have so far and **Run** to let the model create an automated annotation which you can then later edit

i) Repeat previous steps. At the end, you should find the created labels inside the “labels” folder in your Project folder.

B Appendix - Poster



Improving Computer Tomography COVID-19 Lesion Segmentation Using Reproducible Pathways and Data Augmentation Techniques

Introduction

The COVID-19 pandemic has put an immense pressure on the medical system. Testing for COVID-19 is thus crucial for early detection and prevention of future infections.

The current gold standard of **RT-PCR** has a **low sensitivity** and a **high processing** time. RT-PCR testing kits, laboratory personnel and equipment are often not available in large enough quantities.

Alternative: Medical imaging in the form of lung computer tomography (CT) **Lesion Segmentation** and Classification

However, CT scans require professional radiologists to manually annotate each lesion which is **time consuming**.

Solution: employ machine learning techniques such as Deep Learning to **automatically annotate lesions**

Methodology

Data: 329 lung CT scans from 3 different sources (250 of which are from a single source), verified by professional radiologists

Data preprocessing:
Lung segmentation
Extracting the lung from the CT

Deep Learning Models:

❖ **U-Net**
- Convolutional Neural Network
- 23 layers
- Specialized in image segmentation
- Two paths:
  - Contracting – captures the context of the image
  - Expansive – ensures precise localization

❖ **nnU-Net** (Dynamic U-Net)
- Automatically adapts to any dataset (increases generalization)
- Uses 2D, 3D and 3D-Cascade U-Net

Reproducible pathway by using open-source:
- MONAI framework
- data
- preprocessing and data augmentation techniques

Author: Bogdan Palfi
Supervisor: Yeshaswini Nagaraj

Data augmentation techniques:
- **Noise filters: 15%**

- adding random Gaussian noise
- **Sliding windows**

- only pass smaller, consecutive "windows", not the full image
- **Maximum Intensity Projection** (MIP)

- map 3 adjacent slices into one by selecting the maximum intensity

Results

- **U-Net**: sliding windows resulted in the best AUC of 0.8 (F1 = 0.51, sensitivity = 0.6, specificity = 0.99, precision = 0.44)
- **nnU-Net**: noise filters resulted in the best AUC of 0.81 (F1= 0.59, sensitivity = 0.63, specificity = 0.99, precision = 0.59)

No MIP MIP
Predictions given by U-Net and nnU-Net

Discussions

Highest performance: **noise filters** and **sliding windows**

MIP did not aid the segmentation task

The differences between U-Net and nnU-Net are not visible in the current study. However, nnU-Net might have a higher **generalization capability**.

Predictions:
- **Overestimations** – the models might not be fully adapted to the task
- **Underestimations** – comorbidities such as interstitial lung disease or fibrosis could be visible in the lung CT

Limitations:
- Low amount of data (only 329 CT, 30-50 left for testing)
- Low quality data: open-source data often lacks the quality standards of private data, meaning blurrier CT scans and missing annotations for certain visible lesions

Future work:
- MIP for classification tasks, not just segmentation
- Minimum Intensity Projection instead of Maximum Intensity Projection
- More high quality data with the help of professional radiologists and the annotation tool MONAI Label