Maastro | Maastricht UMC+ | TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Clinical Module 2 - 8FM30

Decision making and intervention

Optimisation of deep learning based lung segmentation on chest radiography of COVID-19 positive patients

Group 1

Project coordinator: Dr. Ir. J. Dhont

Student name:	**Student ID:**	**Email address:**
H.J. Damen	1000587	h.j.damen@student.tue.nl
S.H.A. Verschuren	1245272	s.h.a.verschuren@student.tue.nl

Friday 14th May, 2021

Abstract

The COVID-19 pandemic has had a tremendous impact on the hospital workflow, in which each new patient has to be transferred to the radiology department for COVID-19 detection in order to work safely. An optimized workflow using *Smart-Detect* would introduce a mobile CXR scanner near the hospital's entrance for quick scanning. A machine learning-based approach would mark the scanned lungs as potentially pathological or not. In order for *Smart-Detect* to be developed and this model to be trained, a robust method for automated lung segmentation was developed using Generative Adversarial Networks for image-to-image translation of raw CXR images to delineated lung masks. This method proved to be relatively robust for segmenting healthy lungs, as well as lungs with pathology similar to COVID-19-related pneumonia, with an average accuracy of 0.95 ± 0.03 and an average dice score of 0.93 ± 0.06. These findings encourage the adoption of this method in the *Smart-Detect* pipeline.

1 Introduction

As of may 1st, 2021, there have been over 150 million confirmed cases of the 2019 novel Coronavirus Disease (COVID-19) and over 3 million confirmed deaths, as per the World Health Organisation [1]. With so many infected patients, many of whom need treatment, the workload in hospitals and other care centers has increased dramatically. This has resulted in a need to scale down non-COVID-19-related healthcare. For example, cancer diagnoses in the Netherlands were down by 26% compared to pre-COVID-19 numbers. Especially skin cancers were diagnosed more sporadically compared with other years, with only 40% as many diagnoses [2]. This decrease in diagnosed cancers can not be explained by an expected decrease in cancer in the population. Instead, these numbers show a lower priority for non-acute hospital visits.

Besides the impact on the ability of doctors to provide as much care and the willingness of the patients to visit a hospital if their symptoms are mild, the COVID-19 pandemic has also impacted the workflow in hospitals. That is, in order not to accidentally spread the virus in the hospital itself, every patient needs to be transferred to the radiology department, where they are screened for suspicious lung tissue. This first screening requires an X-ray of the thorax (CXR) and a radiologist to inspect this CXR. If nothing suspicious is found by the radiologist and the patient experiences no COVID-19-related symptoms, the patient can go to get the treatment they came for originally. If the CXR shows some irregularities, however, the patient is once again transferred to a Computed Tomography (CT) scanner, where another scan is made. This scan is then inspected by a radiologist, who annotates the CT and may or may not recommend further intervention. Regardless of whether the incoming patient has any pulmonary problems or not, this elongated workflow takes a lot of additional time and resources, as well as the risk of additional COVID-19 infections being slightly increased by having to transfer patients to the radiology department. Given the already increased pressure on the hospital staff and the limited capacity of the hospitals, this is problematic. Hence, it would be desirable to alleviate some pressure by optimizing the workflow, while still working responsibly and safely around COVID-19.

In order to do this, a slim and mobile Dual-Energy Technology Computed Tomography for automatic COVID-19 diagnosis and detection (*Smart-Detect*) is being developed [3]. The goals of the Smart-Detect project are two-fold: to create a compact rapid cone beam CT scanner, capable of single and dual-energy CT scanning to develop machine learning methods (based on Deep Learning) trained on large databases of available COVID-19 CT images to detect disease patterns automatically in 2D and 3D X-ray images. This would introduce a mobile Xray-CT unit that can be positioned near a hospital's entrance and scan every incoming patient near-instantaneously. This would remove the need to transfer every patient to the radiology department. Moreover, the radiologist's interpretation of the scans would be replaced by machine learning software that can detect and identify suspicious tissue.

An important first step in the development of this software is the automatic segmentation of the lungs in a CXR. This is because the goal is specifically to detect abnormal lung tissue and should thus not take surrounding tissues into account. Logically, by removing the surrounding tissue, automatic segmentation of the lungs allows for better training of the proposed Artificial Intelligence (AI) method. This can be done using various approaches, such as deep learning. However, deep learning models have shown some performance issues when segmenting lungs with abnormalities. This is likely due to the large interpatient variance in CXRs, as well as the fact that lesions yield hyperdense regions in the lung tissue, which are relatively less distinguishable from the surrounding tissue [4]. Since this application of this work specifically requires good performance on abnormal lungs, a different approach is needed.

A more recent approach in image analysis is promising for this task: generative adversarial networks (GANs) [5]. This multi-component model consists of two models, a generator and a dis-

criminator. Here, the generator tries to create an image whilst the discriminator tries to discern whether it is from a real dataset or a fictitious one. These can be trained until the generator can create images that are indistinguishable from real images. This approach can also be applied to image-to-image translation, as shown by `pix2pix` [6]. Here, the generator receives an input image and returns an output image that is indistinguishable from the desired delineation of the input image. In the context of lung segmentation, this means that the generator receives a CXR image and returns a binary segmentation mask of the lungs.

The goal of this research is now to train such a GAN model based on the `pix2pix` architecture, thus to create a generator model that can be used for automated segmentation of the lungs. Subsequently, the code was to be designed as comprehensive and easily usable for clinical application.

2 Methods

2.1 Data Acquisition

For training and testing, a combination of three datasets was used. These are the Montgomery county (*MC*) CXR set, the Shenzhen CXR set and a selection from the Valencian Region Medical ImageBank (BIMCV) COVID-19+ set. None of the used datasets contain augmented data.

The *MC* set contains 138 frontal CXRs from *MC*'s Tuberculosis screening (TBC) program, of which 80 are normal cases and 58 are cases with manifestations of TBC. The size of these CXR images is either 4,020×4,892 or 4,892×4,020 pixels [7].
The *Shenzhen* set contains 662 frontal CXRs, of which 326 are normal cases and 336 are cases with manifestations of TBC, including pediatric X-rays. The size of the images in this set can vary but is approximately 3000×3000 pixels [7].
The *BIMCV COVID-19+* set is a large dataset containing CXR and CT imaging of COVID-19 positive patients along with their radiographic findings, pathologies, polymerase chain reaction (PCR), immunoglobulin G (IgG) and immunoglobulin M (IgM) diagnostic antibody tests and radiographic reports [8]. The first iteration of this dataset contained data from 1,311 patients and this number has since increased. However, only a selection of 172 CXR images was used. Note that this selection only contains COVID-19 positive patient data and that this selection process was explicitly based on being particularly challenging for segmentation. This selection makes the total dataset more diverse. A greater data diversity generally improves model performance and has been shown to do so in lung segmentation before [9]. The size of the images in this selection is 512x512 pixels.

The structure of the following section will be similar to the structure of the corresponding code. Please note that each of the described steps will be put in a separate function and in designated function files corresponding to the different stages of this approach.

2.2 Preprocessing

Before the images can be used for training, they are to be preprocessed. Firstly, all images are loaded from the *"raw"* data folder and checked for whether they are accompanied by a mask file. If not, they are omitted from the dataset. Secondly, since the MC image masks are split into a right and left lung mask, these separated masks are combined into a single mask. Thirdly, all images are checked for intensity inversions in the grayscale spectrum, which is caused due to different CXR conventions across countries and institutions. Now, an intensity normalisation is performed, in which every image is normalized to an intensity range of (0, 255). This normalization is performed through a remapping strategy, as is given in equation 1.

$$A_{norm} = (A - min(A)) \cdot \frac{255}{max(A)} \tag{1}$$

with A a 2D matrix representing a grey-scale image.

The fifth and final step is a reshaping step. Here, each image and its corresponding mask are reshaped into a square by padding the sides, after which they are resampled to the final 256x256 images. The now preprocessed images are saved to the *"preprocessed"* folder.

2.3 Dataset generation

To load the now preprocessed images into a dataset usable for model training, the images in the *"preprocessed"* folder are loaded. The 8-bit pixel values are then converted to floats. The input CXR images are standardized to $[\mu = 0;\ \sigma = 1]$ and the masks are binarized to (0, 1). Finally,

the entire combined dataset is split into a training- and test-set. This is an 80-20 split. No data augmentation was performed.

2.4 Model definition

At this stage, the architectures of the generator and discriminator models are defined. The generator is an encoder-decoder architecture whereas the discriminator is a CNN classifier model. Both are 'full image'-based and based on the architectures as described by `pix2pix` [6]. Here, note that the models were implemented in Keras 2.4, using the TensorFlow 2.4 back-end.

For the purpose of denoting the model architectures compactly, let `Ck` denote a Convolution-BatchNorm-ReLU layer with k filters, while `CDk` denotes a Convolution-BatchNorm-Dropout-ReLU layer with a dropout rate of 50%.

For the generator algorithm, the architecture consists of the encoder:
`C64-C128-C256-C512-C512-C512-C512`
and decoder:
`CD512-CD512-CD512-C512-C256-C128-C64`
Hereafter, a convolution is applied to map to the number of output channels (1 in this case), followed by a Tanh function.

The discriminator architecture is:
`C64-C128-C256-C512-C512`
After the last layer, a convolution is applied to map to a 1-dimensional output, followed by a Sigmoid function to map to the final prediction.

As an exception to the previously introduced notation, BatchNorm is not applied to the first `C64` layers of the encoder and discriminator architectures. All ReLUs in the encoder and discriminator are leaky with a slope of 0.2, while the ReLUs in the decoder are not leaky [6].

The loss function is defined by a linear combination of three loss terms.
Firstly, the binary cross entropy (BCE) is given as such:

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i * log(p(y_i)) + (1 - y_i) * log(1 - p(y_i)) \tag{2}$$

with label y and $p(y)$ the predicted probability of y being of the class *"lung"* for each of the N pixels.

Next, the mean absolute error (MAE) is given by:

$$MAE = \frac{\sum_{i=1}^{N} |y_i - x_i|}{N} \tag{3}$$

with prediction y_i and true value x_i for each of the N pixels.

Finally, the Dice score loss (DSL) is given by:

$$DSL = 1 - DSC = 1 - \frac{2 * |X \cap Y|}{|X| + |Y|} \tag{4}$$

with X and Y being the collection of pixels classified as *"lung"* in the prediction and label, respectively. Here, note that the DSL is given by 1 - Dice Score (DSC), so that each element of the loss function should be minimized rather than maximized.

Each of the three loss terms has a weight, which are given by 1, 100, and 100 respectively.
The total loss function is thus as such:

$$Loss = 1 \cdot BCE + 100 \cdot MAE + 100 \cdot DSL \tag{5}$$

The Adam optimizer is used as the optimizer during training. Here, the initial learning rate is set to $2 \cdot 10^{-4}$ and the β_1-hyperparameter is set to 0.5.
The Adam optimizer is given by:

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{6}$$

with η the initial learning rate, $\hat{m}_t$ the exponential average of gradients along weights w_j and $\hat{v}_t$ the exponential average of squares of gradients along weights w_j.

2.5 Training and monitoring

In traditional deep learning, the model receives an input image and predicts some output, which is then compared to the ground truth to calculate and minimize some sort of loss function. This is done iteratively, until the loss is sufficiently minimized. However, since a GAN consists of two competing models, neither loss can be minimized without increasing the other model's loss. Instead, to train the GAN, a slightly more elaborate approach is required. In the following segment, we will elaborate upon the processing steps for a single batch. Please note that figure 1 may assist in providing additional illustration.

Firstly, as can be seen in figure 1A, we generate real samples by simply selecting a batch of random images (X_{realA}) and corresponding masks (X_{realB}) from the training dataset. This data is accompanied by the corresponding label "real" (y_{real}), which will be used for later training of the discriminator.
Then, as is also depicted in figure 1A, we generate fake masks (X_{fakeB}) by running X_{realA} through the generator model. These are then accompanied by the label "fake" (y_{fake}).
The next step is to train the discriminator model on the now acquired samples, which is shown in figure 1B. Firstly, we feed X_{realA} and X_{realB} through the model, after which the predicted label is compared with y_{true} to find the loss. This basically tells the discriminator model what "real" masks for a set of specific images should look like, as the model weights are now updated as a function of the loss. Secondly, we feed X_{realA} and X_{fakeB} along with y_{fake} through the model much in the same way as we did previously. This step teaches the discriminator what the generated/"fake" masks for a set of real images looks like.
Now, it is time to train the generator model, which is depicted in figure 1C. This is done by comparing the previously generated X_{fakeB} to the target mask X_{realB} to find the generator loss, after which the generator model weights are updated accordingly. [6].

For the purpose of monitoring the training, which will be elaborated upon in the next paragraph, the training dataset is split into an "actual" training set and a validation set before the training commences. This split is again 80-20. This yields a total overall split between training, validation and testing of 64-16-20, which is a ratio upheld over the three sub-datasets (*MC*, *Shenzhen* and *BIMCV COVID-19+*). The creation of these datasets enables the calculation of training and validation evaluation metrics at training time. The training monitoring metrics chosen were the generator/discriminator losses, as well as the training and validation MAE.

To stop the training of the model at the appropriate time and thus to prevent under- or overfitting, an early-stopping functionality was built. Here, the validation MAE is logged each epoch. After some moving average filtering to reduce the influence of outliers and stochastic variability,

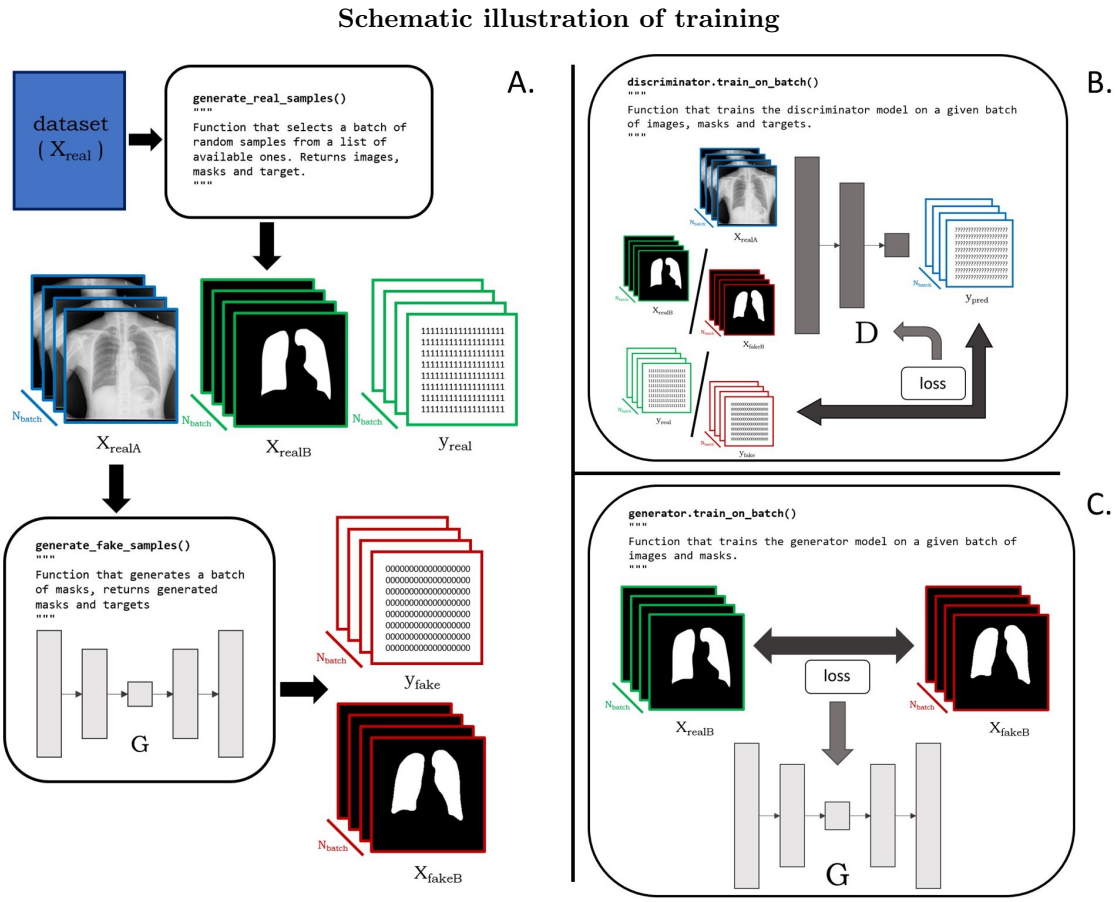

Figure 1: Schematic illustration of the training process. ***A:*** Generation of real and fake samples. ***B:*** Training of the discriminator model. ***C:*** Training of the generator model.

it is checked whether the validation MAE keeps improving with increasing epochs. A *"patience"* parameter is implemented, which is a measure for how many epochs we wait for the model performance to improve. By default, the *patience* is set to 100 epochs due to the unstable nature of the training process of this model, while the maximum number of epochs is set to 1000 epochs. As the model is stored after each epoch, the model weights may be reset to the model yielding the best loss.

2.6 Post-processing

After training the model, the generator model may be used to generate lung masks for input CXR images. In some cases, however, segmentation is not perfect. For example, there could be some "holes" in the mask (false negative), or some false positive "blobs" on the sides of the images, as will be demonstrated in the next section. This has to do with the nature of this machine-learning based approach, as simple conditions such as *"there should likely be two lungs"* or *"lung masks generally do not contain holes"* are not prescribed. To handle the absence of these prescriptions, which would generally be the case in non-machine learning approaches such as morphology and thresholding, some postprocessing is performed. The process will be elaborated upon shortly in the next segment.

Firstly, one should note that the generator output is approximately, but not exactly, binary. The first postprocessing step should thus be to convert this raw output into a binary mask. This is

simply done by applying a threshold of 0.5. Some other thresholds were experimented with, but this did not make a big difference due to the near-binary nature of the raw output mask.
Next, a morphological opening was performed on the binary mask. Here, a disk-shaped structuring element with a diameter of 10 pixels was used. Here, please note that the mask shape is still 256x256, and thus constant for all images. This opening is performed to disentangle the lungs from any non-lung blobs in the segmentation mask so that afterwards, these may be removed.
For the next step, the two largest uninterrupted areas (in theory being the two lungs) are selected from the mask. The other positive pixels are regarded as false positive and omitted.
Now, a final morphological closing of the mask with a disk-shaped structuring element with a diameter of 10 pixels is performed to fill in any gaps in the final lung mask.
Finally, if applicable, the mask images are resampled and then cropped to the original CXR image size.

2.7 Evaluation

To review the performance of the model, a comparison can be made between the model-predicted and ground truth masks. For the purpose of comparison, two metrics were selected, being the accuracy (eq. 7) and DSC (eq. 4) metrics.

$$accuracy = \frac{N_{TP} + N_{TN}}{N_{TP} + N_{FP} + N_{TN} + N_{FN}} \tag{7}$$

with N_{TP}, N_{FP}, N_{TN} and N_{FN} being the number of *true; false positive* and *true; false negative* pixels, respectively.

Please note that the final evaluation of the model was naturally performed on the test dataset. For a more complete representation of the model's performance, the evaluation metrics were calculated for the entire test-dataset, as well as for each individual test-dataset (*Shenzhen, MC, BIMCV COVID-19+*).

2.8 Model application

For easy application of the model created in this research, a separate jupyter notebook file (`www.jupyter.org`) has been created to apply the trained generator model on new input images. A number of different functions can be called upon. By default, this notebook requires the paths to the previously trained model, an *in* folder and an *out* folder. Here, it simply runs through the entire processing pipeline and saves masks for the CXR images stored in the *in* folder to the *out* folder. Please note that the notebook file was designed to be easily adaptable to whatever the desired application in practice may be.

3 Results

This section will elaborate upon the results achieved in this research. Key points will include inspection of the training process, inspection of the postprocessing and finally qualitative and quantitative assessment of the final results.

3.1 Training

Figure 2: Smoothed Mean Absolute Error curves for the model during training. Note that the training MAE is given in pink, while the validation MAE is given in green. On the x-axis, the iteration number is given ($n_{iter} = n_{epoch} \cdot N_{images}$).

Figure 2 shows the in-training MAE curves. The pink line shows the training MAE, whereas the green line represents the validation MAE. Please note that these curves were smoothed for readability purposes. After 342 epochs, training was stopped by the early stopping criterion described in subsection 2.5 in order to prevent overfitting. Training exited with a validation MAE of 0.0232. Figure 2 shows a decrease in training and validation loss until epoch 342.

3.2 Postprocessing

Predicted masks before and after post-processing

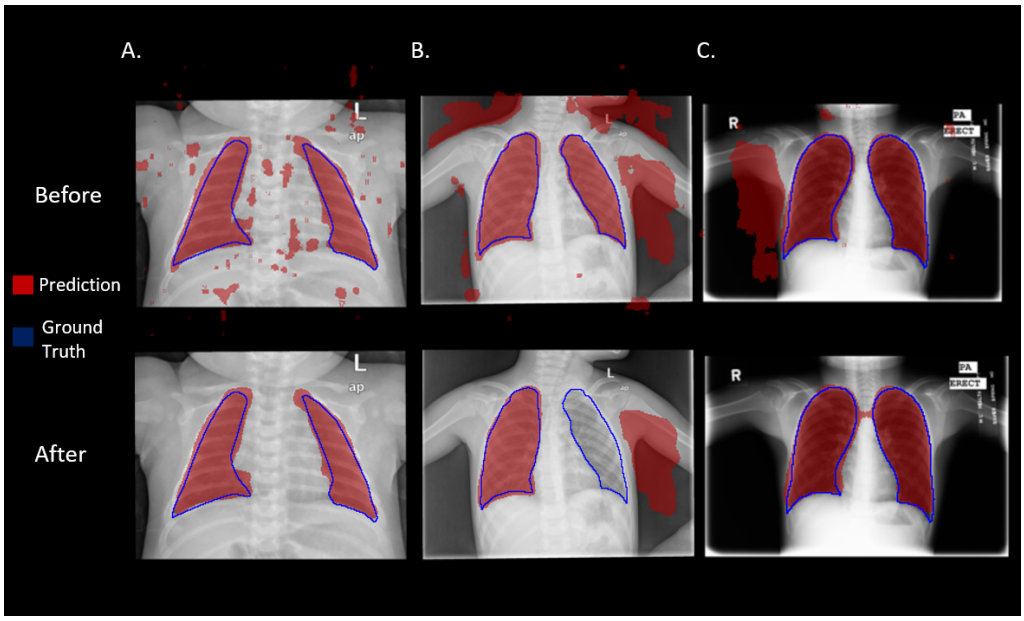

Figure 3: Model-predicted outputs before and after post-processing, cherry-picked from the data for illustrative purposes. Here, please note that the ground truth mask is given by the blue contour, while the model predictions are given by the red areas. The top row contains predictions before postprocessing, while the bottom row illustrates predictions after postprocessing. The columns represent subjects A, B and C, respectively.

Figure 3 shows the input images with the model-predicted outputs and the expert delineated masks of three non-pathological, but otherwise challenging CXRs. These images were cherry-picked from the data for the purpose of illustrating the postprocessing process. Here, note that subject A is a paediatric CXR, while subjects B and C are CXRs from a "further away" perspective than is generally the case for the training dataset. As can be seen from figure 3, the expert delineated areas generally overlap more with the images after post-processing. Subjects B and C, however, also illustrate some shortcomings of the postprocessing process. For subject B, it can be seen that the left lung is misplaced due to a major segmentation artefact below the left arm, while for subject C the left and right lung masks have been "connected".

3.3 Evaluation

Here, a qualitative and quantitative evaluation of our proposed method will be given. Please note that for the qualitative analysis, a subdivision will be made between non-pathological lungs and pathological lungs to ensure deeper understanding of this method's performance. For the qualitative analysis, the evaluation metrics proposed in subsection 2.7 will be given for each dataset, after which notable results will be elaborated upon.

Non-pathological lungs

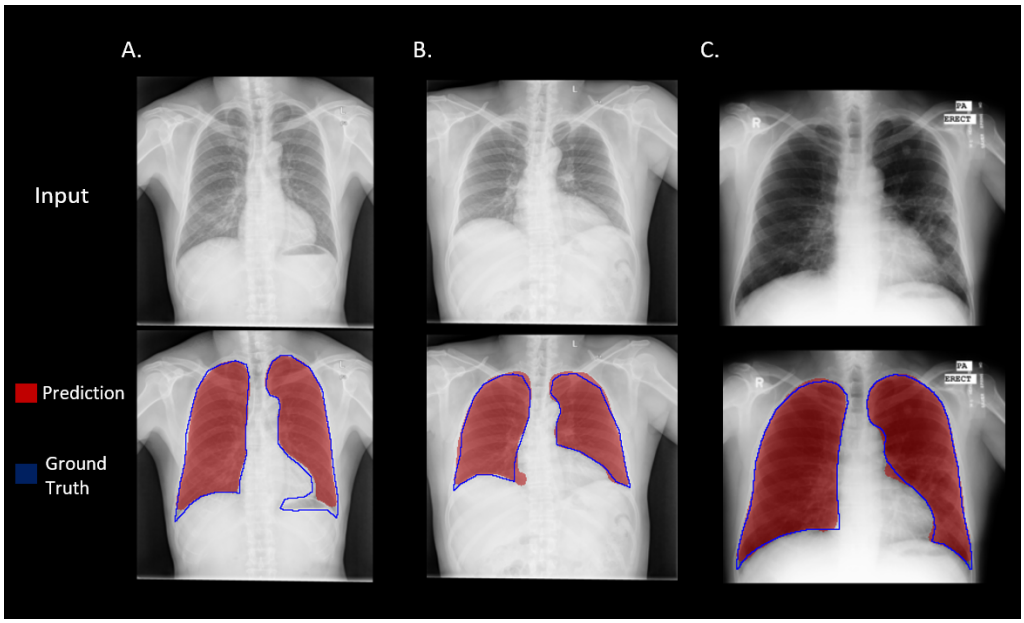

Figure 4: Predicted and expert-delineated lung masks of non-/minimally pathological CXR images. Here, please note that the ground truth mask is given by the blue contour, while the model predictions are given by the red areas. The columns represent subjects A, B and C, respectively.

Figure 4 shows the input images, the model-predicted outputs, and the expert delineated masks of three non-pathological CXRs. In general, it may be noted that the prediction overlaps quite nicely with ground truth. For subject A, a part of the left lung just below the heart is false negative. Also, for subject B, it may be noted that the medial bottom part of the right lung is false positive.

Pathological lungs

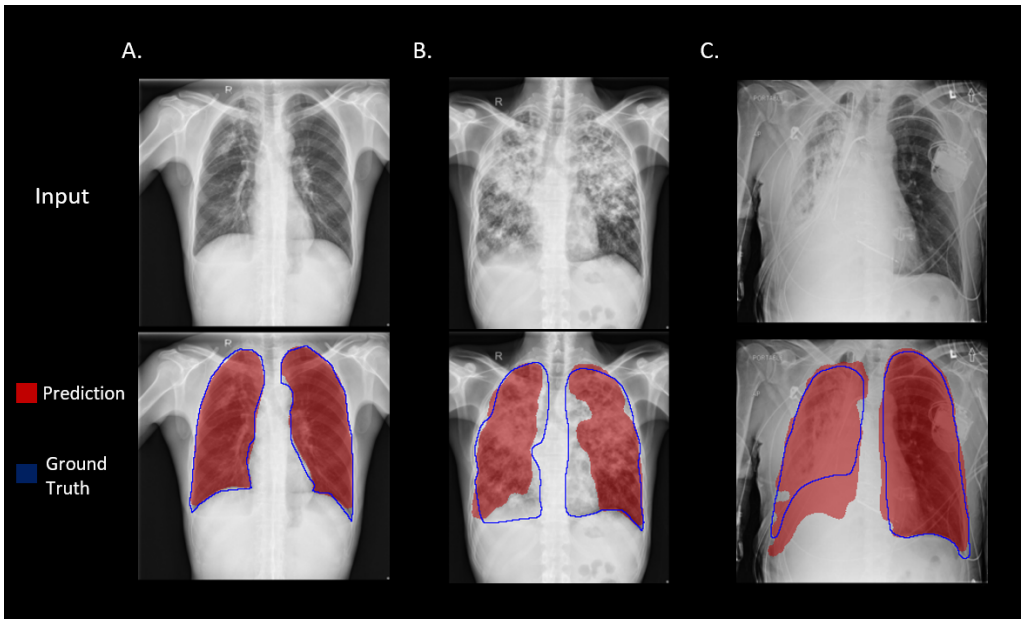

Figure 5: Predicted and expert-delineated lung masks of pathological CXR images. Here, please note that the ground truth mask is given by the blue contour, while the model predictions are given by the red areas. The columns represent subjects A, B and C, respectively.

Figure 5 shows the input images, the model-predicted outputs, and the expert delineated masks of three pathological CXRs. For subject A, relatively small hyperdense regions are found at the medial parts of the lungs. Here, our method's prediction coincides quite nicely with the expert delineation. Subject B shows relatively severe hyperdense regions all over both lungs. Here, the prediction matches less with ground truth due to some false negative predictions at the bottom of the right lung and the medial left lung. For subject C, a severe hyperdense region is found all over the right lung, as well as a pacemaker superpositioned on the left lung. Here, it may be noted that the pacemaker does not interfere with our method's prediction. Additionally, the prediction of the right lung's location seems to somewhat overlap with ground truth, apart from some false positive regions at the bottom of the lung.

Table 1: Evaluation metrics for the final, postprocessed test data

Dataset	**Accuracy** $(\mu \pm \sigma)$	**DSC** $(\mu \pm \sigma)$	**n**
Shenzhen	0.96 ± 0.01	0.94 ± 0.05	123
MC	0.97 ± 0.01	0.96 ± 0.02	20
BIMCV COVID-19+	0.90 ± 0.05	0.87 ± 0.06	31
All	0.95 ± 0.03	0.93 ± 0.06	174

As was discussed in subsection 2.7, the overall averaged results can be summarized by the average accuracy and the average DSC of an independent test dataset. As becomes apparent from table 1, the average accuracy of the proposed method is 0.95, while the average DSC equals 0.93. These metrics were also calculated for the three separate datasets. Here, it becomes apparent that the accuracy and DSC are lower for the *BIMCV COVID-19+* set compared to the *Shenzhen* and *MC* sets, while the standard deviation (i.e. spread) of these values is relatively higher for the *BIMCV COVID-19+* dataset.

4 Discussion

For the purpose of improving automatic CXR-based lung pathology detection, as is being developed by *Smart-Detect*, we explored the possibilities of GAN-based lung segmentation using image-to-image translation. This approach was implemented on the *Shenzhen*, *MC* and *BIMCV COVID-19+* datasets. In the next section, we will discuss some noticeable results and implications of this research, as well as how our approach relates to relevant literature.

In subsection 3.2, some illustrative results were given for the postprocessing process. Here, it was shown that in figure 3B, the postprocessing failed due to a rather large artefact below the left arm. This occurred due to the fact that the gaps between the arms and chest of the subject were classified as lung tissue. The reason the postprocessing failed is because the artefact was in fact bigger than one of the lungs, which caused the algorithm to classify the artefact as being the left lung and omits the actual lung. In figure 3C, a problem emerged in which the two lung masks were "connected" in the postprocessing process. This is due to the final closing step of the mask having too large of a structuring element for this image. Both of these issues emerge from the fact that these images were from a "farther away" perspective than the rest of the dataset. To alleviate these problems in the future, an algorithm that crops the image to a more appropriate size could be created.

In subsection 3.3, some qualitative and quantitative analysis was shown. In the qualitative analysis part (figure 4), it became apparent that our method performs quite well on non-pathological lungs. Based on our research purpose, however, it is of vital importance that our method is relatively robust for pathological lungs. These results, given in figure 5, implied rather good robustness against mildly hyperintense regions (figure 5A) as one would expect to see with COVID-related pneumonia. In case of severe fibrosis (figure 5B) or a unilateral severely hyperintense lung (figure 5C), our method proved slightly less robust, yet these cases would be rare in our use case and a high performance here seems slightly less important for our specific purpose.
From the quantitative analysis, which was based on evaluation metrics defined in subsection 2.7 and presented in table 1, it becomes apparent that overall, our method yields an accuracy of 0.95 ± 0.03 and a DSC of 0.93 ± 0.06, based on a total test dataset size of 174 subjects. When analyzing the three dataset upon which our combined dataset is built, it becomes apparent that the *BIMCV COVID-19+* dataset yields a relatively low accuracy and DSC of 0.90 and 0.87, respectively. This could be explained by the fact that this dataset was specifically selected for the purpose of being difficult to perform segmentation on. Each image contains COVID-19-related pneumonia or other abnormalities. Moreover, in this specific dataset, it seems that different conventions are upheld regarding image acquisition than for the other two datasets. This manifests itself as intensity inversions and a seemingly lower X-ray penetration, leading to poor contrast.

In literature regarding lung segmentation, there are various different methods for the automated segmentation of lung tissue. However, not all methods have been applied on the same datasets. This is relevant, as this research has specifically targeted the issue of segmenting lungs with abnormalities. For example, the Japanese Society of Radiological Technology (JSRT) dataset contains pathological lungs, but does not contain any deformations that affect lung shape and deformation [4]. Hence, a certain performance on JSRT need not be representative of the performance on the *MC*, *Shenzhen* or *BIMCV COVID-19+* sets.

An interesting deep learning variant on conventional deep learning approaches found in the literature which has been proven to perform relatively well on deformed lung segmentation is *image reconstruction* [10]. Here, lung segmentation is performed twice, where a first segmentation performs well on non-deformed lungs. The second segmentation is specifically targeted at the situations where the first segmentation was not as accurate. These are the situations with abnormal lung tissue, where the contrast between regions is affected and texture features are therefore

unreliable. A second trained CNN reconstructs the abnormal lung regions to mitigate this issue. This approach was used on the *MC* set and achieved an average accuracy of 0.97 as well as a DSC of 0.94. The results of our method on the *MC* dataset are marginally better, being 0.97 and 0.96 for accuracy and DSC, respectively.

Thus, image-to-image translation of CXR images using GANs seems to be a robust method for the automated segmentation of healthy lungs, as well as lungs with pathology similar to COVID-19-related pneumonia. The average accuracy of our proposed method is 0.95 ± 0.03 and the DSC equals 0.93 ± 0.06. Although some other deep learning approaches can achieve higher scores, many of these approaches are untested on datasets with CXR images affected in shape and texture. Our proposed GAN approach performs very similarly, if not marginally better than recent image reconstruction methods tested on the same dataset [10].

5 Conclusion

In this work, an attempt was made to assist in automatic CXR-based lung pathology detection, as is being developed by *Smart-Detect*, by exploring the possibilities of GAN-based lung segmentation using image-to-image translation. The results show our method to be robust for the automated segmentation of healthy lungs, as well as lungs with pathology similar to COVID-19-related pneumonia, yielding an accuracy and DSC of 0.95 ± 0.03 and 0.93 ± 0.06, respectively. These results include an accuracy of 0.90 ± 0.05 and DSC of 0.87 ± 0.06 on severely affected lungs. These findings encourage the adoption of our method into the *Smart-Detect* pipeline, having implemented some of our suggestions for future work regarding accuracy and robustness improvement.

References

[1] World Health Organisation. WHO Coronavirus (COVID-19) Dashboard. https://covid19.who.int/ (accessed: 2021-05-11).

[2] Avinash G. Dinmohamed, Otto Visser, Rob H.A. Verhoeven, Marieke W.J. Louwman, Francien H. van Nederveen, Stefan M. Willems, Matthias A.W. Merkx, Valery E.P.P. Lemmens, Iris D. Nagtegaal, and Sabine Siesling. Fewer cancer diagnoses during the COVID-19 epidemic in the Netherlands. *The Lancet Oncology*, 2020.

[3] Health Holland. Novel imaging device for rapid detection of covid-19 and other lung diseases. https://www.health-holland.com/project/2020/2020/novel-imaging-device-rapid-detection-covid-19-and-other-lung-diseases (accessed: 2021-05-11).

[4] Sema Candemir and Sameer Antani. A review on lung boundary detection in chest X-rays. *International Journal of Computer Assisted Radiology and Surgery*, 14(4):563–576, apr 2019.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, oct 2020.

[6] Phillip Isola, Jun Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pages 5967–5976. Institute of Electrical and Electronics Engineers Inc., nov 2017.

[7] Stefan Jaeger, Sema Candemir, Sameer Antani, Yì-Xiáng J Wáng, Pu-Xuan Lu, and George Thoma. Two public chest X-ray datasets for computer-aided screening of pulmonary diseases. *Quantitative imaging in medicine and surgery*, 4(6):475–477, 2014.

[8] Maria de la Iglesia Vayá, Jose Manuel Saborit, Joaquim Angel Montell, Antonio Pertusa, Aurelia Bustos, Miguel Cazorla, Joaquin Galant, Xavier Barber, Domingo Orozco-Beltrán, Francisco García-García, Marisa Caparrós, Germán González, and Jose María Salinas. BIMCV COVID-19+: a large annotated dataset of RX and CT images from COVID-19 patients. *arXiv*, jun 2020.

[9] Johannes Hofmanninger, Forian Prayer, Jeanny Pan, Sebastian Röhrich, Helmut Prosch, and Georg Langs. Automatic lung segmentation in routine imaging is primarily a data diversity problem, not a methodology problem. *European Radiology Experimental*, 2020.

[10] Johnatan Carvalho Souza, João Otávio Bandeira Diniz, Jonnison Lima Ferreira, Giovanni Lucca França da Silva, Aristófanes Corrêa Silva, and Anselmo Cardoso de Paiva. An automatic method for lung segmentation and reconstruction in chest X-ray using deep neural networks. *Computer Methods and Programs in Biomedicine*, 177:285–296, aug 2019.