Lung Segmentation Challenge: Deep Learning Approach

Group Number:

Helena Bartels, Ian McBride, Pengcheng Ren, Sicheng Zhao, Sidharth Mehra, Tianxia Song,

August 15, 2022

Abstract

Chronic respiratory diseases pose a major public health problem, with an estimated 3.9 million deaths in 2017, accounting for 7% of all deaths worldwide. This makes the problem of identifying chest-related complication very much severe. Deep learning and AI techniques have potential to reduce the radiologist workloads and can help in a timely diagnosis of these respiratory diseases. In this group work, we applied the U-Net model for segmenting chest X-ray images obtained from two publicly available datasets. We achieved an accuracy of 95% with the pre-trained model. This report serves as an initial case study to explore the potential of deep learning in assisting the routine radiological workflows.

1 Introduction

Image identification tasks in machine learning can broadly be divided into three main categories; classification, segmentation and detection. In image classification, the model identifies what is in the image for a given object (such as a cat or a dog), in image detection the model identifies where in the image the object is with a bounding box around the object, while in a segmentation task each individual pixel belonging to the object is identified. Hence image segmentation, where automatic detection of boundaries within images is performed, is a more advanced task compared to image classification or detection. In this report, we train a U-Net model to segment human lungs based on data obtained from two open source datasets of chest X-rays.

The chest X-ray is the most common radiological test performed worldwide and accounts for 25% of all diagnostic imaging techniques [3][7]. It allows for an evaluation of the airways, mediastinum, the heart and pleura and chest wall. The indications for performing a chest X-ray are broad and include for asymptomatic people undergoing health screening, as part of investigations for sepsis even in the absence of respiratory symptoms, detecting respiratory disease and for patients who are critically unwell or medically unstable [8]. Hence the chest X-ray is a frequently performed diagnostic investigation which can provide useful information for many clinical conditions, at a relatively low cost and with minimal risk to the patient per examination [1]. As a result, two datasets, known as the Montgomery County and Shenzhen chest X-ray sets [4], were made available by the United States Library of Medicine in an effort to provide data for the scientific community for training machine learning models on chest X-rays.

The Montgomery County chest X-ray set was obtained in collaboration with the department of health and human services in Montgomery county in the United States [4]. All chest X-rays were obtained as part of the counties tuberculosis screening programme and consists of 138 frontal chest X-rays. Of these, 80 are normal studies and 58 have manifestations of tuberculosis. A clinical reading of the chest-X ray is included for each case. The second dataset, the Shenzhen X-ray set, was obtained through collaboration with Shenzhen People's hospital in Shenzhen, China [4]. Chest X-rays were performed over a one month period as part of routine outpatient clinics. The dataset consists of 662 frontal chest X-rays, of which 326 have no pathology (normal) and 336 have features of tuberculosis. As for the Montgomery dataset, ground truth clinical labelling is available for all images.

Given the volume of chest X-rays performed on a daily basis in both community and hospital care, clinicians are faced with a high volume of images to review (6). Furthermore, conditions such as tuberculosis, which is a significant global health burden, are screened for using chest X-rays, hence efforts to utilise machine learning and artificial intelligence (AI) in the triage and diagnose of chest X-rays have emerged [5][2] . Results to date have been promising with AI algorithms outperforming radiologists in the detection of conditions such as tuberculosis [5]. For many of these studies, an initial task is segmenting the lungs. A collaboration between the Radiological Society of North America (RSNA) and Kaggle, as well as the US National Institute of Heath and MD.ai developed the RSNA Pneumonia detection challenge. The challenge was to build an algorithm for the detection of pneumonia from the above mentioned open source datasets. In this paper, we use a Kaggle notebook developed as part of the RSNA pneumonia challenge to train a U-net lung segmentation task.

2 Methodology

2.1 Architecture

U-Net is a segmentation technique originally proposed for medical imaging segmentation in a 2015 paper [6]. The model architecture is straightforward: an encoder, for down sampling, and a decoder, for up sampling, with skip connections. As Figure 1 shows, it shapes like the letter U hence the name U-Net. The grey arrows represent the skip connections that concatenate the encoder feature map with the decoder, this assists in the backward flow of gradients and improves training.

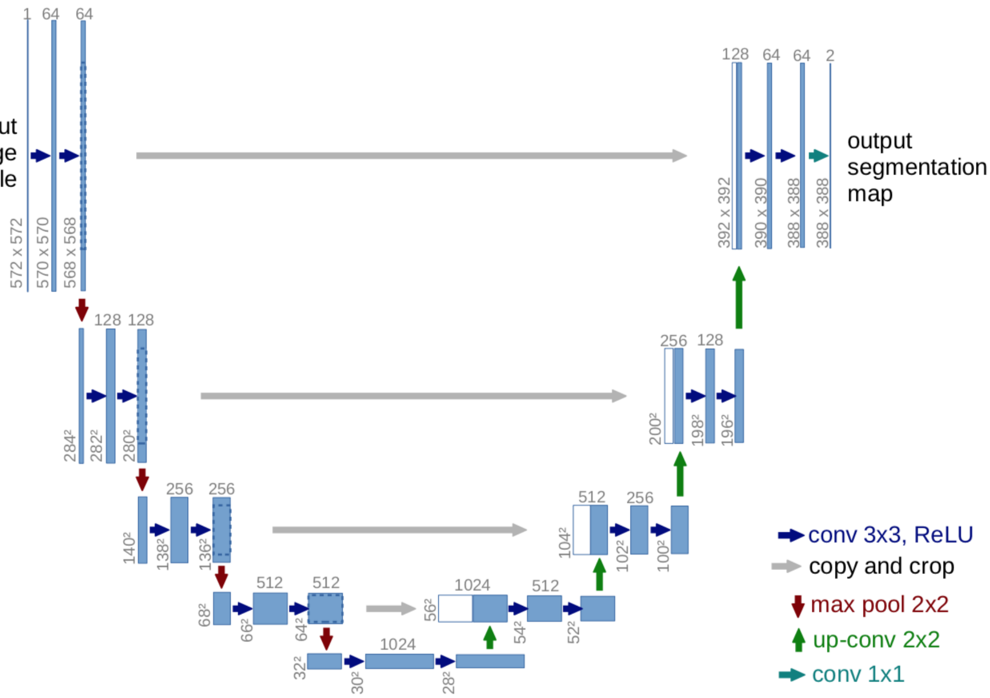

Figure 1: The original U-Net architecture

2.2 Defining the model

Three iterative blocks will be used in defining the U-Net model, namely the convolution operation block, the encoder block, and the decoder block. With the help of these three blocks, the U-Net architecture can be constructed with ease. The convolution operation block is used to perform the primary operation of taking the entered input parameters and processing a layer of convolution operations. Arguments, the input for the convolution layer, the number of filters and padding, are included. The value of padding as "same" is used to maintain the same shape of the convolutional layers. The encoder architecture block will use consecutive inputs starting from the first layer all the way to the bottom. The encoder function is a defined in the convolutional block. Once passed through the convolution blocks, these elements are quickly downsized. The decoder block will include three arguments, namely the receiving inputs, the input of the skip connection, and the number of filters in that particular building block. The entered input will be up sampled in the model. The receiving input and the newly up sampled layers will then be concatenated to receive the final value of the skip connections. This combined function will then be used to perform a convolutional block operation to proceed to the next layer and return this output value.

```
(1, 512, 512, 1)
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
ImageInput (InputLayer)         (None, 512, 512, 1)  0
__________________________________________________________________________________________________
gaussian_noise_2 (GaussianNoise (None, 512, 512, 1)  0           ImageInput[0][0]
__________________________________________________________________________________________________
UnetEncoder (Model)             [(None, 512, 512, 8) 24388       gaussian_noise_2[0][0]
__________________________________________________________________________________________________
conv2d_17 (Conv2D)              (None, 32, 32, 32)   9248        UnetEncoder[1][4]
__________________________________________________________________________________________________
leaky_re_lu_17 (LeakyReLU)      (None, 32, 32, 32)   0           conv2d_17[0][0]
__________________________________________________________________________________________________
up_sampling2d_5 (UpSampling2D)  (None, 64, 64, 32)   0           leaky_re_lu_17[0][0]
__________________________________________________________________________________________________
concatenate_5 (Concatenate)     (None, 64, 64, 64)   0           up_sampling2d_5[0][0]
                                                                 UnetEncoder[1][3]
__________________________________________________________________________________________________
conv2d_18 (Conv2D)              (None, 64, 64, 32)   18464       concatenate_5[0][0]
__________________________________________________________________________________________________
leaky_re_lu_18 (LeakyReLU)      (None, 64, 64, 32)   0           conv2d_18[0][0]
__________________________________________________________________________________________________
up_sampling2d_6 (UpSampling2D)  (None, 128, 128, 32) 0           leaky_re_lu_18[0][0]
__________________________________________________________________________________________________
concatenate_6 (Concatenate)     (None, 128, 128, 64) 0           up_sampling2d_6[0][0]
                                                                 UnetEncoder[1][2]
__________________________________________________________________________________________________
conv2d_19 (Conv2D)              (None, 128, 128, 32) 18464       concatenate_6[0][0]
__________________________________________________________________________________________________
leaky_re_lu_19 (LeakyReLU)      (None, 128, 128, 32) 0           conv2d_19[0][0]
__________________________________________________________________________________________________
up_sampling2d_7 (UpSampling2D)  (None, 256, 256, 32) 0           leaky_re_lu_19[0][0]
__________________________________________________________________________________________________
concatenate_7 (Concatenate)     (None, 256, 256, 48) 0           up_sampling2d_7[0][0]
                                                                 UnetEncoder[1][1]
__________________________________________________________________________________________________
conv2d_20 (Conv2D)              (None, 256, 256, 32) 13856       concatenate_7[0][0]
__________________________________________________________________________________________________
leaky_re_lu_20 (LeakyReLU)      (None, 256, 256, 32) 0           conv2d_20[0][0]
__________________________________________________________________________________________________
conv2d_21 (Conv2D)              (None, 256, 256, 32) 9248        leaky_re_lu_20[0][0]
__________________________________________________________________________________________________
leaky_re_lu_21 (LeakyReLU)      (None, 256, 256, 32) 0           conv2d_21[0][0]
__________________________________________________________________________________________________
up_sampling2d_8 (UpSampling2D)  (None, 512, 512, 32) 0           leaky_re_lu_21[0][0]
__________________________________________________________________________________________________
concatenate_8 (Concatenate)     (None, 512, 512, 40) 0           up_sampling2d_8[0][0]
                                                                 UnetEncoder[1][0]
__________________________________________________________________________________________________
conv2d_22 (Conv2D)              (None, 512, 512, 16) 5776        concatenate_8[0][0]
__________________________________________________________________________________________________
leaky_re_lu_22 (LeakyReLU)      (None, 512, 512, 16) 0           conv2d_22[0][0]
__________________________________________________________________________________________________
lungs (Conv2D)                  (None, 512, 512, 1)  145         leaky_re_lu_22[0][0]
__________________________________________________________________________________________________
cropping2d_2 (Cropping2D)       (None, 416, 416, 1)  0           lungs[0][0]
__________________________________________________________________________________________________
zero_padding2d_2 (ZeroPadding2D (None, 512, 512, 1)  0           cropping2d_2[0][0]
==================================================================================================
Total params: 99,589
Trainable params: 99,587
Non-trainable params: 2
__________________________________________________________________________________________________
```

Figure 2: The original U-Net architecture

2.3 Data Split

Data augmentation is the process of modifying, or "augmenting" a dataset with additional data. This helps to increase the performance of the model by improved generalising and thereby reducing overfitting. Augmentation can be applied as "pre-processing" before training or as "real time" during training. Other simple augmentation techniques include cropping, sheering, zooming in or out, and adjusting brightness or contrast. In this model, horizontal flip, rotation, shift, shear, and zoom are all utilised in real time using ImageDataGenerator() from keras.

2.4 Loss function

Neural networks may be designed to produce a continuous range of output values while others require a 1/0 output. Mean Squared Error (MSE) loss is ideal for the first type of task, for the second type, a classification task, a different type of loss function would be ideal. Binary Cross Entropy (BCE) loss is suited for a classification task, it penalises wrong but confident outputs in addition to correct but not so confident outputs. BCE is used in this model as image segmentation consists of the classification of each pixel in an image.

2.5 Optimiser

The technique for back propagating gradients to update network weights has several options available. Adam was used in this case. It utilises the idea of momentum to reduce the chances of getting stuck in a local minimum. It also uses a separate learning rate for each learnable parameter, which can adapt to how each parameter changes during training. An alternative would be Stochastic Gradient Descent (SGD), it is popular due to its simplicity, and it is lightweight in terms of computational resources however it can get stuck in local minima in the loss function and a single learning rate applies to all parameters.

2.6 Controlling the learning process

A hyperparameter is a parameter whose value is used to control the learning process. The learning rate is an example of a hyperparameter. In addition to altering the learning rate with the optimiser, a scheduler can decrease the learning rate if there has not been a reduction in validation loss in set number of epochs. For this model this number, known as the patience, was set to three. The maximum number of epochs is another hyperparameter that can be altered, in this model it was set to 25. An early stop trigger was set so that if the validation loss has not decreased in 15 epochs the model would terminate.

3 Results

The pre-trained model from Xu hao, 2019 is used here, and we have only partially fine-tuned it. Figure 3 shows the results on the training and validation sets.

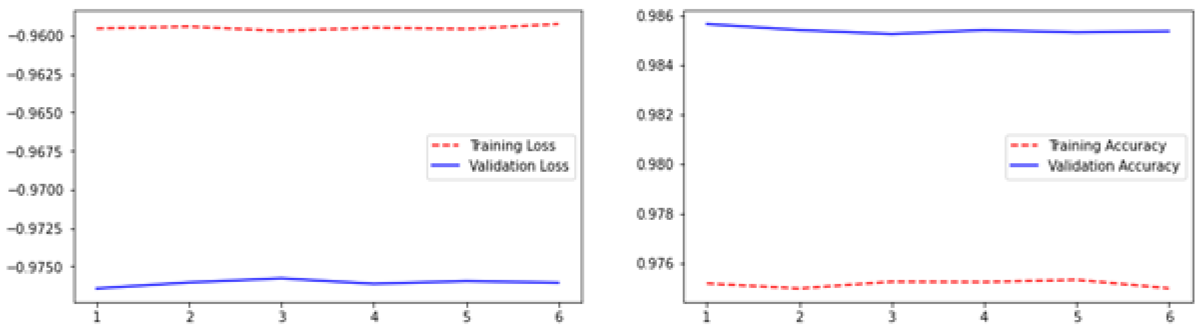

Figure 3: Training/Validation Loss and MIOU Vs Number of Epoch

Since the original model has shown amazing results and early stopping is set here, the convergence is completed in just six Epochs. Although there is a large gap between the training set and the validation set shown in the training images. However, in terms of values, the difference between the two is only between 0.01. It can be tentatively judged that no overfitting is produced. In addition to that 97-98% accuracy is amazing and shows the performance of the model very well.

In the test set, we use the MIOU metric to measure the performance of the model. MIOU is a very common metric in image segmentation. It is represented in mathematics as the intersection-to-merge ratio. Compared to accuracy, the intersection ratio is more focused on the effectiveness of the segmented region. It is relatively more indicative of the model's strengths and weaknesses. In the MIOU metric, the MIOU value of the model in the test set is 0.95308775, with no significant decline in performance relative to the training set.

In the test set, we use the MIOU metric to measure the performance of the model. MIOU is a very common metric in image segmentation. It is represented in mathematics as the intersection-to-merge ratio. Compared to accuracy, the intersection ratio is more focused on the effectiveness of the segmented region. It is relatively more indicative of the model's strengths and weaknesses. In the MIOU metric, the MIOU value of the model in the test set is 0.95308775, with no significant decline in performance relative to the training set. The following images show the performance in specific images. A side-by-side comparison is done for mask and prediction to get a better intuition of the model's performance.

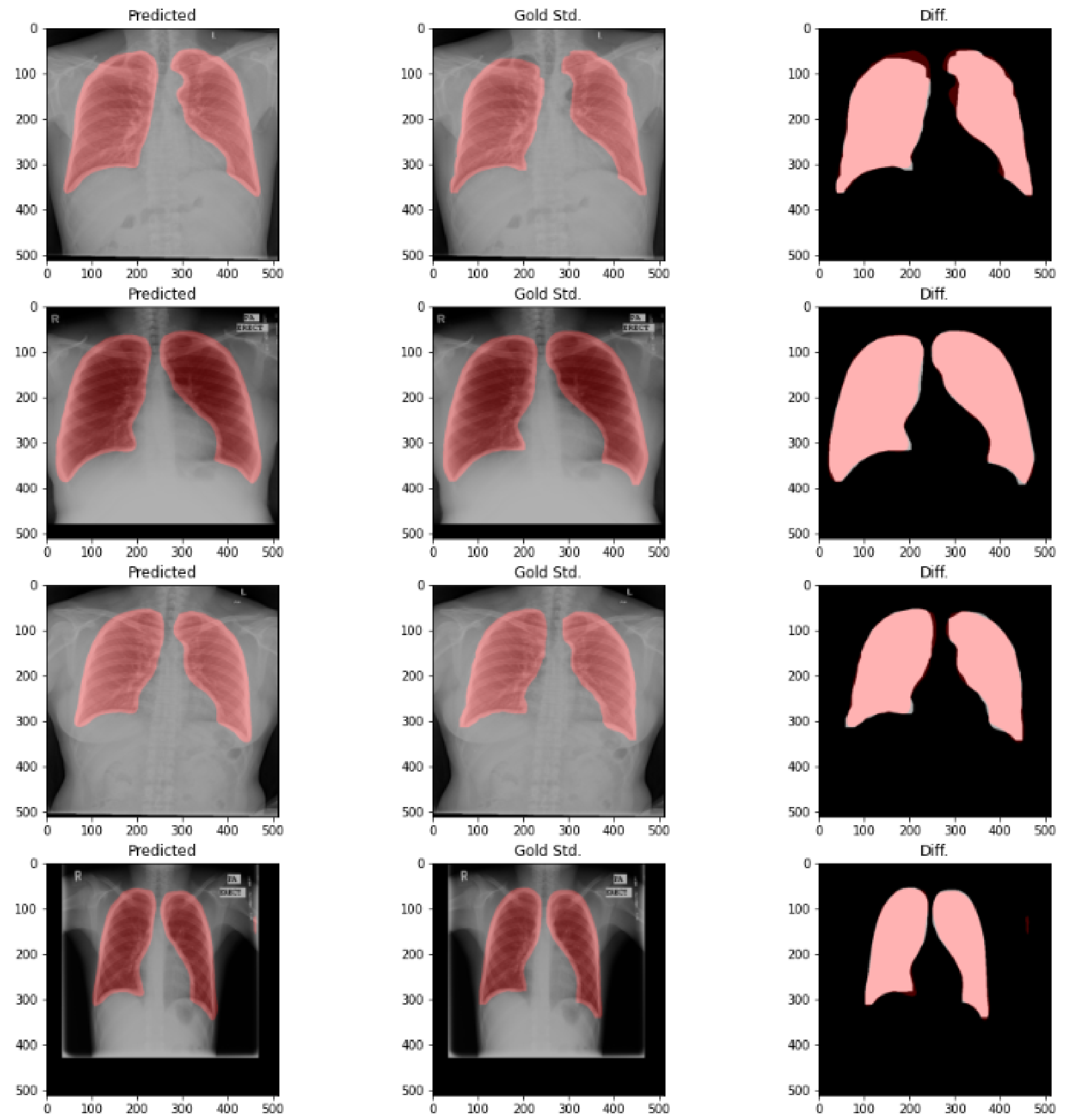

Figure 4: Side-by-side of mask, ground truth and difference for a selection of images

4 Discussion

Mask-rcnn follows the idea of Faster RCNN, with a ResNet-FPN architecture for feature extraction and a Mask prediction branch for predicting binary masks. It not only detects targets in the image, but also gives a high quality segmentation result for each target, and can be extended to other tasks such as key point detection. Mask-rcnn has the advantage of powerful network feature extraction, excellent target detection and fine segmentation of instances.

Mask-rcnn is more suitable for semantic recognition of large data, while unet is more suitable for semantic segmentation with simple structure and small data sets (e.g. cell segmentation, MR and CT segmentation, etc.). The segmentation ability of mask-r-cnn has to be tested on larger datasets. unet is surprisingly strong and performs well on very small training sets.

One important reason why the task of medical image segmentation has not been solved satisfactorily until today is the complexity and diversity of medical images. Due to the differences in imaging principles of medical images and the properties of human tissues, as well as the formation of images affected by noise, field shift effects, local body effects and tissue motion, medical images are inevitably blurred and in homogeneous compared with ordinary images. In addition, the anatomical structure and shape of the human body is complex and varies considerably from person to person. All these make the medical image segmentation difficult.

5 Future Work

Although most depth segmentation models for medical image analysis rely solely on clinical images for prediction, there is often multi modal patient data in the form of other imaging modalities as well as patient metadata that can provide valuable information, which is not used by most depth segmentation models. Therefore, a valuable research direction to improve the segmentation performance of medical images is to develop models that can utilize multi-modal patient data. Medical image segmentation technology has made some breakthroughs, but it also faces the following problems:

- Lack of high-quality data. Deep neural networks typically require a large number of annotated examples to perform training tasks. In medical image processing, collecting large case annotation data sets is often a very difficult task.
- Due to the different image reconstruction methods and medical imaging equipment, it may lead to inconsistent offset field and uneven gray level `enumerate` environment.
- The medical image data with different imaging principles can only reflect the specific information of the human body, but cannot reflect the comprehensive information characteristics.

Based on the above problems, the development direction of deep learning medical image segmentation technology mainly focuses on the following points:

1. Expanding image data through data enhancement; Transfer learning method was used to combine pre-training of large data sets and fine-tuning of target data sets. The weakly supervised learning method effectively combines the advantages of unsupervised pre-training and supervised learning.

2. Batch normalization, regularization and Dropout can be used to improve the gray level inhomogeneity. `enumerate` environment.

3. To improve the accuracy of analysis by integrating multimodal medical images based on the complementary information between different images.

With the further improvement of computer technology and the continuous optimization and innovation of deep learning algorithms, medical image segmentation technology based on deep learning has great development potential, which will be more widely used in various fields of medical research and have a more profound impact.

References

[1] Stephen H Bradley, Nathaniel Luke Fielding Hatton, Rehima Aslam, Bobby Bhartia, Matthew EJ Callister, Martyn PT Kennedy, Luke TA Mounce, Bethany Shinkins, William T Hamilton, and Richard D Neal. Estimating lung cancer risk from chest x-ray and symptoms: a prospective cohort study. *British Journal of General Practice*, 71(705):e280–e286, 2021.

[2] Erdi Çallı, Ecem Sogancioglu, Bram van Ginneken, Kicky G van Leeuwen, and Keelin Murphy. Deep learning for chest x-ray analysis: A survey. *Medical Image Analysis*, 72:102125, 2021.

[3] Sacha F de Stoppelaar, Liza Pereverzeva, Bram Hafkamp, Nikki Lips, Floor Tielbeke, Linde Rustenburg, Caroline Hoogerheide-Wiegerinck, and Koen de Heer. Diagnostic value of chest x-ray in patients with suspected infection and no respiratory signs or symptoms. In *Open Forum Infectious Diseases*, volume 7, page ofaa221. Oxford University Press US, 2020.

[4] 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, 2014.

[5] Zhi Zhen Qin, Shahriar Ahmed, Mohammad Shahnewaz Sarker, Kishor Paul, Ahammad Shafiq Sikder Adel, Tasneem Naheyan, Rachael Barrett, Sayera Banu, and Jacob Creswell. Tuberculosis detection from chest x-rays for triaging in a high tuberculosis-burden setting: an evaluation of five artificial intelligence algorithms. *The Lancet Digital Health*, 3(9):e543–e554, 2021.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[7] F Shannoun. Medical exposure assessment: the global approach of the united nations scientific committee on the effects of atomic radiation. *Radiation protection dosimetry*, 165(1-4):125–128, 2015.

[8] Anouk M Speets, Yolanda van der Graaf, Arno W Hoes, Sandra Kalmijn, Alfred PE Sachs, Matthieu JCM Rutten, Jan Willem C Gratama, Alexander D Montauban van Swijndregt, and Willem PThM Mali. Chest radiography in general practice: indications, diagnostic yield and consequences for patient management. *British Journal of General Practice*, 56(529):574–578, 2006.