A Look into U-Net: Explaining the Most Revolutionizing Image Segmentation Algorithm

André Oskar Andersen (wpr684)
wpr684@alumni.ku.dk

Abstract

This work aims at explaining the U-Net, introduced by Ronneberger, Fischer and Brox (Ronneberger et al., 2015). The network is trained on the Shenzhen chest X-ray dataset (Jaeger et al., 2014). Throughout this paper we first conclude, that the model is affected by the bias in the training data, due to an imbalanced class distribution. Secondly, we conclude, that the goal of the bottleneck of the model is to create a latent space, where samples are placed, such that semantically similar sampels are placed close to each other. We further conclude, that (1) the edges of the lungs in a given input image are the most important features of the image, (2) if the input image is of a woman, then the bottom part of the chest image is especially important, and (3) for people with abnormality in the lungs, this abnormality can also be very important. Lastly, we conclude, that (1) the segmentation up until the very last convolution layer of the model, works by predicting the boundary between the lungs and everything else, (2) the job of the encoder and the majority of the bottleneck is to encode the data in the latent space, and (3) the last step of the bottleneck comes with a rought estimation of the lung-boundary, which is then further adjusted as the input is moved through the decoder of the model.

1 Introduction

Medical image segmentation is one of the most common use cases for machine learning, as the machine learning model can help with detecting diseases or accurately apply radiography treatment to a patient. However, with these models it is not only important that they yield very accurate results, but also that we understand them, such that we know their strengths and weaknesses.

One of the most common and revolutionizing algorithms for medical image segmentation is U-Net, developed by Ronneberger, Fischer and Brox in 2015 (Ronneberger et al., 2015). Likewise, one of the most common types of medical image segmentation tasks is the segmentation of the lungs of a chest X-ray image.

The aim of this paper is thus to implement and train U-Net on a dataset of chest X-ray images for lung segmentation, as well as to explain the developed U-Net to get an understanding of how the model works, as well as to get an understanding of why it delivers suboptimal performance for some samples.

2 The Dataset

	Size	Ratio
Training set	340	.6
Validation set	113	.2
Testing set	113	.2
Total	566	1

Table 1: Data distribution.

For the data we use the Shenzhen chest X-ray dataset (Jaeger et al., 2014). The dataset consists of 662 chest X-ray images, where most of these images have a corresponding binary segmentation mask. For each data sample the dataset also includes information about the gender and age of the person in the X-ray image, as well as whether or not any abnormality is seen in the lungs of the person.

All of the images consists of 4 channels. The sizes of the images varies, but they do all have a height and width of about 3000 pixels. We decided to cast each image to grayscale and reduce its height and width to 512 pixels to speed up the training-process. One could argue, that by doing so we discard a lot of important information, which could lower the performance of the algorithm. However, a 512×512 will still capture

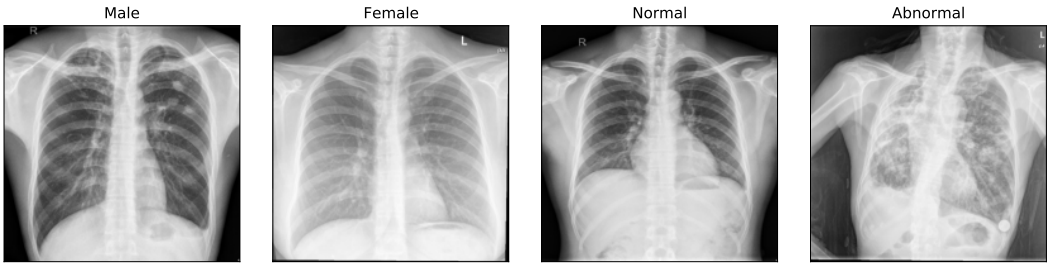

Figure 1: Examples of four processed chest X-ray images.

a lot of the information, so we mostly discard the unnecessary finer details of the images. Figure 1 illustrates some of the resulting X-ray images.

For the samples that do not have a corresponding segmentation mask we simply discard them, as we will be doing supervised binary segmentation, where these samples cannot contribute with anything, resulting in the dataset consisting of 566 samples.

Lastly, we split the dataset into three non-overlapping datasplits, as this help us getting an unbiased evaluation of the trained models. Table 1 illustrates the resulting data distribution.

3 U-Net

For the model of choice we will be using U-net, developed by Ronneberger, Fischer and Brox in 2015 (Ronneberger et al., 2015). We chose this model because (1) it has previously been the state-of-the-art model for medical image segmenation, (2) many of the current state-of-the-art models for medical image segmentation are variants of U-Net, and (3) the model has a latent space, which we maybe able to use to get an understanding of how the model works.

For the architecture of our U-Net we try to follow the architecture of the U-Net presented by Ronneberger, Fischer and Brox as close as possible. However, due to memory limits we had to halve the number of filters used by each convolution layer. By doing so the accuracy of our model will probably decrease a bit compared to the original U-Net, however, we do believe that it will still perform very well.

Figure 2 visualizes the architecture of the developed U-Net. In the visualization, the black arrows represent a 3×3 convolution layer followed by a ReLU and a round of batch normalization, the red arrows represent a round of applying a 2×2 max-pooling, the green arrows represent a 2×2 transposed convolution layer for upscaling the input followed by a ReLU, the blue arrows represent the act of concatenating the data at the tail of the arrow to the data at the head of the arrow, and the purple arrow represent a 1×1 convolution layer. Each box represents the produced feature maps from the preceding arrow.

The architecture is split into three parts: (1) the encoder, (2) the bottleneck, and (3) the decoder. The encoder and decoder consists of three layers, where the encoder downsamples the input and the decoder upsamples the input. The bottleneck does not upsample nor downsample the input, but instead just processed the input. The input to the network is an image with just one channel.

The first layer of the encoder returns a processed version of the input image with 32 channels and the following layers in the encoder doubles the amount of channels, such that the input to the bottleneck has 256 channels. Further, the bottleneck also doubles the amount of channels, such that the input to the decoder has 512 channels. Each layer in the decoder then halves the amount of channels, such that the input to the very last convolution layer has 32 layers. Lastly, this very last convolution layer returns an image with just 1 channel.

3.1 Training Details

The model is fitted for 30 epochs by using the Adam-optimizer with an initial learning rate of 0.01 and a batch-size of 8. For the loss-function we use `BCELossWithLogits` provided by PyTorch, where we set the `pos_weight`-parameter to 2.972132932484872, which we got by dividing the number of background pixels in the training

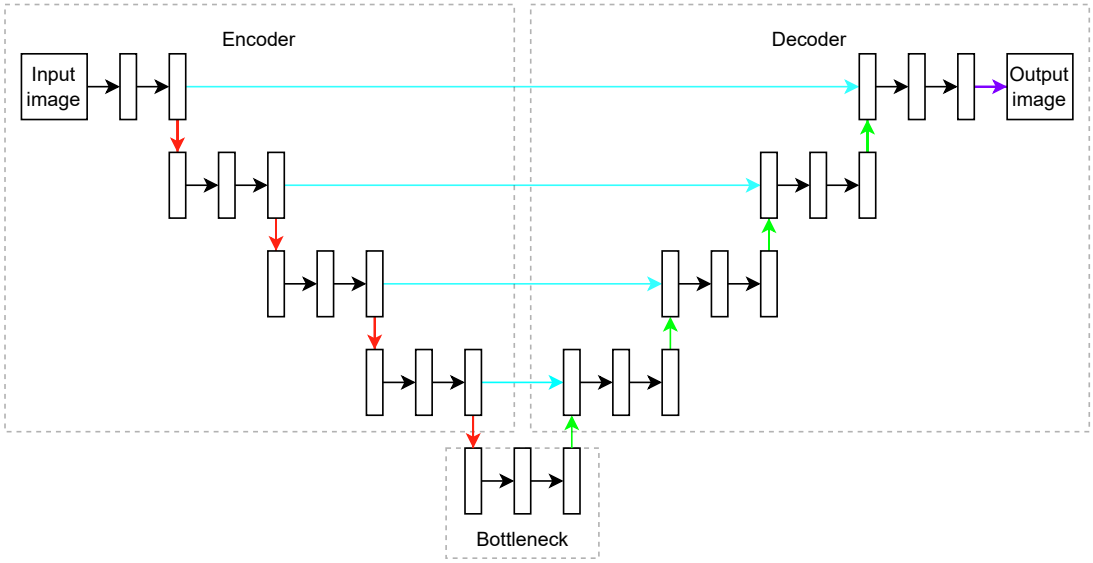

Figure 2: Architecture of our developed U-Net

dataset with the number of foreground pixels in the training dataset. During the training of the model we keep track of the setting of the model that yields the best *IOU*-score on the validation-set. If this IOU-score has not been beaten for 10 continuous epochs the learning rate is dropped with a factor of 0.5. After the 30 epochs the setting of the model that yielded the best IOU-score on the validation-set will be the one that we will be using forward.

4 Results

We have in Figure 3 illustrated the evolution of the training and validation loss, as well as the training and validation IOU-accuracy. By looking at the losses we can see, that both the training loss and the validation loss had plateaued, hence why it would not make sense to keep on fitting the model.

The setting of the model that delivered the best results on the validation data was the model from the 12th epoch. This setting had a training IOU-accuracy of 0.903 and a validation IOU-accuracy of 0.893. When we evaluate this setting of the model on the testing dataset we get an IOU-accuracy of 0.897. For comparison we also implemented a simple baseline model that just predicted the mean training segmentation mask. This model had a training IOU-accuracy of 0.675, a validation IOU-accuracy of 0.673, and a testing IOU-accuracy of 0.676. If we compare our two models against each other we see, that our U-Net completely outperforms the baseline, hinting towards us correctly implementing the model.

	Male	**Female**	**Total**
Normal	187	92	279
Abnormal	202	85	287
Total	389	177	566

Table 2: Distribution of the various classes in the dataset

5 Discovering Biases in the Dataset

Before we explain the developed model, we will be looking at the distribution of the various classes in the dataset to see if it contains any bias, which could be embedded in the developed model.

Looking at Table 2 we can see, that there are more than twice the amount of X-ray images of males, than there are of females, leading to the dataset having a bias towards male patients. Since the dataset is of X-ray chest images and females generally have a bigger chest than males, which is visible in the X-ray images, this gender-bias could be embedded in the model.

On the other hand, the distribution of the images that contains any abnormality is much more balanced, as about 50.7% of the images does not contain any abnormality and the remaining 49.3% of the images does contain some abnormality. This leads us to believe, that the dataset does not contain any bias towards people with or without any abnormality in the chest.

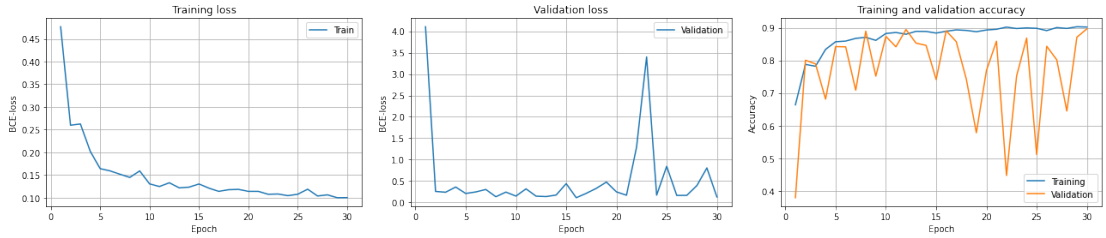

Figure 3: Left: Training loss. Middle: Validation loss. Right: Training and validation IOU-accuracy.

	Male	**Female**	**Mean***
Normal	0.897	0.879	0.891
Abnormal	0.907	0.909	0.908
Mean*	0.902	0.894	

Table 3: IOU-accuracy of the model on the various classes of the dataset. *: The mean takes the class-size into account, such that the bigger classes are weighted higher than the smaller classes.

6 Explaining U-Net

Throughout the following section we will be explaining the implemented U-Net. Due to the very similar performance of the model on the train, validation and test data, the explanation of the model will be happening on all three datasplits as a whole.

6.1 Performance on Imbalanced Dataset

Figure 4: Age group size and running average of the IOU of each age group of the implemented U-Net.

One easy way to check if the imbalanced classes in the data has been embedded in the model is to see if the model delivers the same performance on the various classes. This test cannot conclude, that the model does not have the data bias embedded in the model, but it can conclude that the model has the data bias embedded it in.

We compute the IOU-accuracy of the model on the various classes of the dataset. Table 3 illustrates the IOU-accuracies of the model on the classes of the dataset. Looking at this we can see, that the model delivers very similar results on all of the classes, regardless of the class distribution. Thus, based on this we can conclude, that there are not enough evidence, that the model carries any gender-bias or abnormality-bias.

We further check if any dataset bias is embedding in the model by looking at the performance of the model on the various age group of the dataset. Figure 4 illustrates the size of each age group and the performance of the model on each age group. Looking at the graph we can see, that the two graphs correlates with each other, such that the model performs better on an age group the bigger it is. However, we have to note, that (1) for all age groups the model still outperforms the baseline model and (2) the difference between the performance of the model on the best-performing age group and the worst-age group is not very big.

Overall, there are not any evidence, that the model carries any gender-bias or abnormality-bias. However, there is evidence, that the model performs better on age groups with many data samples than on age groups with fewer data samples. However, the model does still deliver useful predictions for all age groups.

6.2 Clustering the Latent Space

We further explain the model by looking at the structure of the latent space of the model. By "latent space" we refer to the lowest-dimensional space that the model encodes the input data to. In our case, where we use 512×512 images with one channel, each image is encoded to having the dimensions 32×32 with 256 channels in the latent space.

To analyse the latent space we start by flat-

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Male/normal	8 (.06)	64 (0.45)	64 (.49)	51 (.34)
Male/abnormal	93 (.65)	34 (.24)	15 (.12)	59 (.39)
Female/normal	7 (.05)	28 (.2)	44 (.34)	13 (.09)
Female/abnormal	34 (.24)	15 (.11)	7 (.05)	29 (.19)

Table 4: Statistics of the content of the four clusters in the latent space of the model. The numbers outside the parentheses denotes the frequency of the class in the cluster. The numbers inside the parentheses denotes the ratio of the class in the cluster.

tening each sample in the latent space, such that they become a vector of 262.144 elements. Then, we apply K-Means on these samples to see if this yields any interesting results. We chose to use $K = 4$ as there are 4 classes in the dataset.

Table 4 describes the content of the four clusters of the latent space. Looking at the table we can see, that for the first cluster, the majority of the samples belongs to either males with abnormality or females with abnormality, whereas the third cluster mostly consists of samples of males without abnormality or females without abnormality. Based on these two clusters we can conclude, that the model has tried to learn to distinguish between people with and without abnormality, however, these clusters do include some samples that are not supposed to be there, which possibly could describe the suboptimal performance of the model.

For the second and fourth clusters we can see, that they both focuses on samples of male patients, as the two biggest classes in both clusters are of males with and without any abnormalities. Thus, the model does not have any clusters that focuses on samples of female patients, hinting towards the model actually having embedded the gender-bias of the dataset.

Based on this we conclude, that the goal of the bottleneck of the U-Net is to create a latent space, where samples are placed, such that the closer they are to each other in the latent space, the more sematically similar they are in reality.

6.3 The Occlusion Test

Next, we explain the model by finding out what features of the input image are the most important. This is done by performing the *Occlusion test*, where we iteratively occlude each 14×14 region in the image with random noise. Thus, by looking at the loss between the resulting predicted segmentation and the ground truth segmentation mask, we can get a feeling of what features of the input image are the most important (Surma, 2021).

We perform the occlusion test on eight images - the images that yielded the best and worst results for the "male"-class, the "female"-class, the "normal"-class, and the "abnormal"-class. For the loss-function we use the same function as used during training of the model.

Figure 5 illustrates the results of the eight occlusion tests. Looking at the figure we can see, that for the cases where the model performed the best, the most important features of the images were the edges of the lungs. This is probably due to the fact, that none of these images have any outlying features, making just the edges of the lungs the most important feature.

For the the cases where the model performed the worst, the edges of the lungs are never the most important features. For the "male"-class and "normal"-class, the input image was the same. In these cases the most important feature seems to be the pixels that are close the heart of the person. It is difficult to tell why the model find these pixels so important, however, it could be, that (1) the image is slighty rotated, making the relative placement of the heart different than what the image is used to, or (2) the image is of a young child, where the relative placement of the lungs and the heart is different than what the model is used to.

For the "female"-class we see, that the most important features seems to be the pixels at the lower part of the lungs. This could be due to the shape of the patient's breasts being visible in the image. We have already seen, how the model carries some gender-bias, which could support this theory.

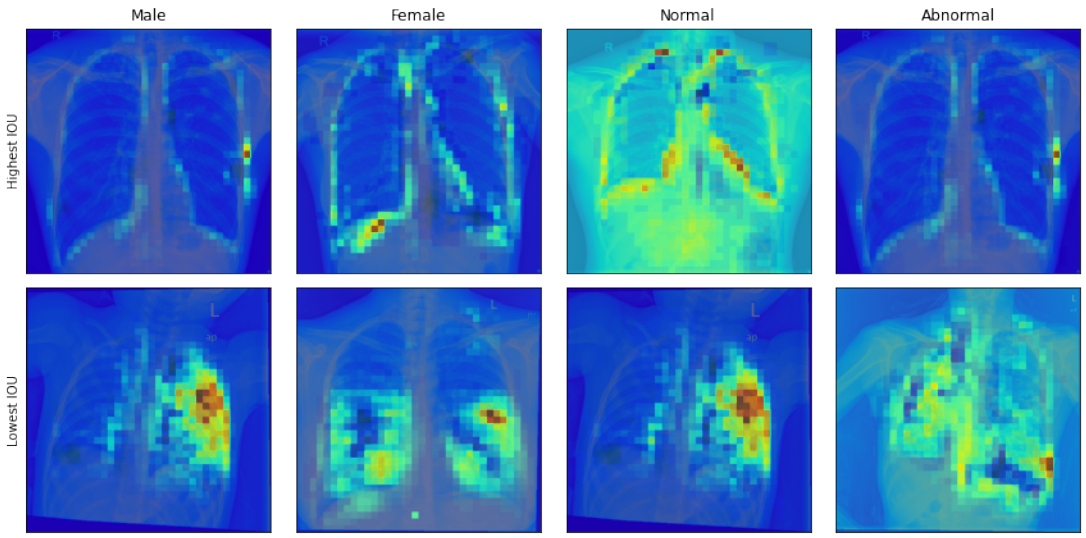

Figure 5: Results of the occlusion test overlayed on top of the input image.

For the "abnormal"-class we see, that the most important features are the pixels in the left lung (from our perspective), as well as the pixels at the bottom of the right lung (from our perspective). This is most likely due to the fact, that the patient has some abnormalities, that heavily changes the size and shape of the lungs, making it hard for the model to predict the edges of the lungs without actually seeing the edges.

Based on these observations we conclude, that the edges of the lungs are the most important features of the images. A rotated input can potentially influence the results, as well as the fact, that the patient may be very young. For the "female"-class, the bottom part of the lungs are also important, since their chest can make it difficult for the model to predict the edges of the lungs. For people with abnormality in the lungs, the abnormality is also important, in cases where the abnormality is very visible.

6.4 Visualizing Feaure Maps

Lastly, we explain the model by looking at the produced feature map of each encoder and decoder layer, as well as the produced feature map of each double convolution operation in the bottleneck. By doing this we can get an idea of the learned tasks of the encoder, bottleneck and decoder of the model. We do this for the eight input images that were used for occlusion test in Figure 5 to see if we can find any further reasoning behind the different performance among the images.

The amount of feature map that is produced by the various steps ranges from 32 to 512. As we cannot visualize and examen every single one of these feature map, we will instead be looking at the average feature map for each of these steps. Further, as the input image is processed by the convolution layers in the network, the height and width of the image is reduced from 512 pixels down to a minimum of 32 pixels, making it difficult to see what is happening in these feature maps. Thus, for visualizing the feature maps, we scale the produced feature maps to have a width and height of 256 pixels, by using `skimage.transform.resize`.

Figure 6 illustrates the nine mean feature maps produced by the model, the final predicted mask, the ground truth mask, as well as the input image, for each of the eight input images. Note, that we have not illustrated the feature map of the very final convolution layer, as this feature map is simply the final prediction of the model.

By looking at Figure 6 we can get an understanding of how the model works. For our first observation we have, that the segmentation up until the very last convolution layer (whose feature map we have not visualized) actually just consists of detecting the boundary between the lungs and everything else. This can explain our observation from the occlusion test, where we conluded, that

Figure 6: Visualization of the mean predicted feature map of each layer of the developed U-net for various inputs.

the edges of lungs are the most important features of the images, as without these edges it is difficult to estimate this boundary. Thus, the task of the very last convolution layer is to use this boundary to classify every pixel as either being part of the lungs or not.

For our second observation we have, that the segmentation of the lungs does not happen before the last step in the bottleneck. Up until that point, the task of the convolution layers is simply to encode the input data in the latent space, such that similar data are placed close to each other in the latent space, as we saw in Section 6.2. Thus, the task of the decoder and the first two steps of the bottleneck is to simply encode and place the data in the latent space.

For our third observation we have, that the last step in the bottleneck comes with a rough estimation of the boundary between the lungs and everything else. This boundary is then adjusted throughout the encoder, where the adjustments are gradually more fine, the further up the encoder the data gets. Thus, the task of the encoder is to estimate and fine adjust the estimation of the boundary between the lungs and everything else.

Lastly, there does not seem to be any differences in the patterns of the feature maps among the different classes (male, female, abnormal, normal), nor between the worst and best predictions.

7 Discussion and Future Work

If we were to work further with this project, it would be ideal to confirm the embedding of the dataset bias in the developed model by (1) finding a similar dataset that is better balanced than the dataset we use, or (2) by making use of techniques such as data agumentation to upsample the less frequent classes. By doing so the dataset should carry less bias, which we can use to see if this results in a less biased model. Further, some of the age groups in the dataset consisted of very few samples, resulting in the evaluation in Section 6.1 of the model on these age groups potentially being very inaccurate. Thus, by using one of the two techniques for gathering more data, the we would get a more accurate evaluation.

Secondly it would be ideal to look at the learned features in finer details. In Section 6.4 we looked at the mean feature map of various parts of the developed model. It could be interesting to look at the individual feature maps instead to see what they have learned. By doing so we could potentially reduce the size of the model if some of the produced feature maps are always very similar. Further, it would be interesting to follow the work of Zeiler and Fergus (Zeiler and Fergus, 2013) by training a backward looking network simultaneous while training the original network. This backward looking network can then be used for better visualizing the learned features of the network.

8 Conclusion

Throughout this paper we have successfully implemented the U-Net by Ronneberger, Fischer and Brox (Ronneberger et al., 2015). We have further explored the network to get an understanding of how it works, as well as observed, that a biased dataset, due to unbalanced class distribution, has resulted in suboptimal performance of the model.

References

Stefan Jaeger, Sema Candemir, sameer Antani, Yì-Xiáng J. Wáng, Pu-Xuan Lu, and George Thoma. 2014. Two public chest x-ray datasets for computer-aided screening of pulmonary diseases.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation.

Greg Surma. 2021. Cnn explainer - interpreting convolutional neural networks (1/n).

Matthew D Zeiler and Rob Fergus. 2013. Visualizing and understanding convolutional networks.