



# AI in medical imaging

Cas van den Boogaard



In modern medicine, imaging devices are widely used to look at the inner workings of our bodies. The field of computer-aided diagnosis (CAD) is concerned with the systems that help doctors interpret the output of those devices. CAD systems can aid in many ways. They can visually improve images (e.g., denoising, balancing brightness), mark regions of interest (e.g., highlight possible tumors), analyze regions of interest (e.g., calculating bone density) or classify an image or region of interest (e.g., a tumor is benign or malignant).

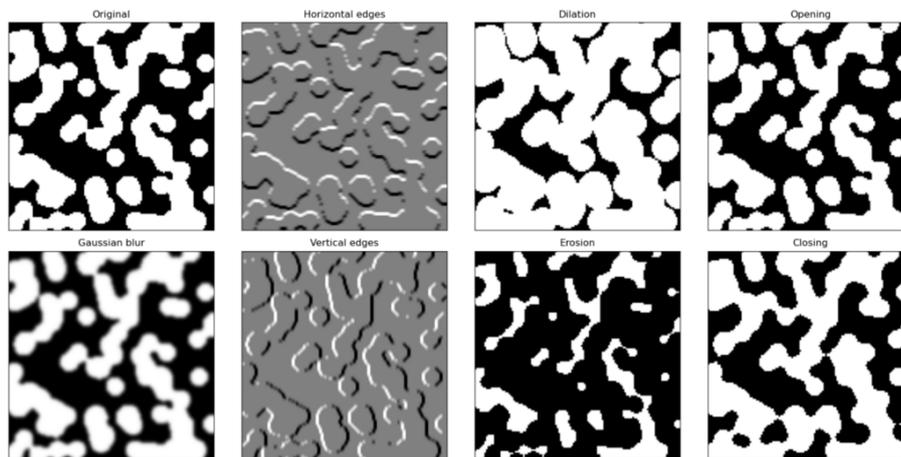
It is only a small step to *computer vision* (teaching computers to understand image data), which has been an exciting topic in the last decade. Artificial intelligence (AI) models have shown state-of-the-art performance on many tasks, such as object detection and pose estimation. Naturally, this progress has also found its way to the medical domain. Even though these AI models provide a new way of analyzing medical image data, many of the underlying mathematical concepts are shared with the 'old fashioned' image processing techniques.

## From image processing to AI

Digital images can be seen as a matrix of numbers that correspond to the pixels in the image. In the case of a black-and-white image, each pixel (and as such each element in the matrix) has a single number that represents the brightness. During image processing, the original image is transformed into a new one. In practice, that means applying different mathematical operations to the image matrix. One widely used group of image processing techniques is called filtering, in which a new value is calculated for each pixel based on the surrounding pixels.

One type of filter is the convolutional filter. Convolutional filtering is done by sliding a second, smaller matrix — the convolution kernel — across the image matrix. In each position, the two matrices are multiplied and summed, resulting in a new value. Changing the values in the convolution kernels changes what task the kernels are suited for, such as blurring an image or detecting edges. Check out [this great tool](#) to experiment with different convolution kernels and to visualize the math that's happening behind the scenes.

There is also morphological filtering, which is the non-linear cousin of convolutional filtering. Again, the neighborhood for each pixel is considered, but a non-linear function is applied to the pixel values instead of doing a multiplication with the convolution kernel. An example is the erosion filter, which you get by taking minimum of all pixel values. The figure on the next page shows the effect of some of these filters.



The effects of some convolution and morphological filters.

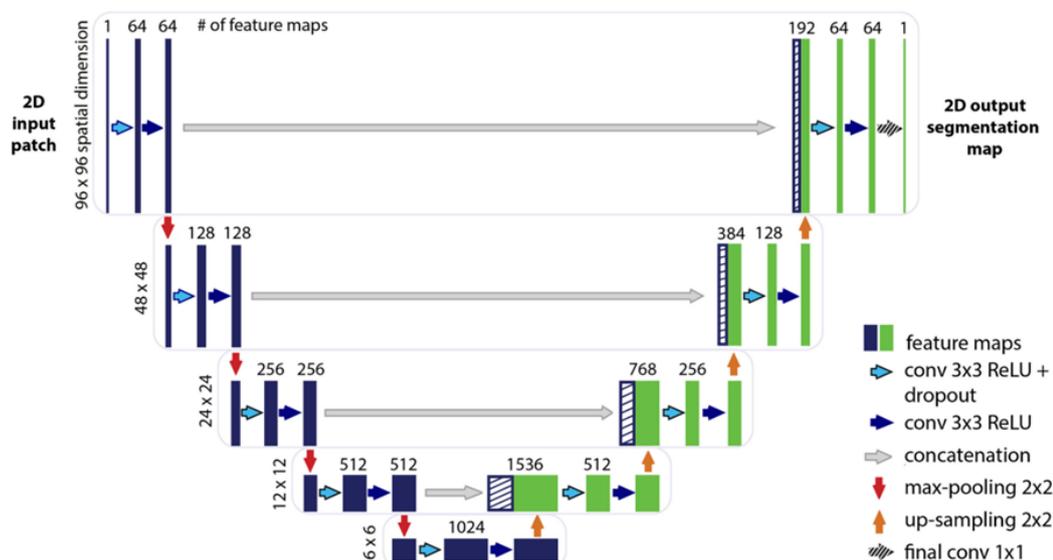
Modern AI models rely heavily on convolution filters. These so-called convolutional neural networks (CNNs) have one or multiple convolutional layers, each of which has a set of convolution kernels. Each of these kernels filters the input image and transforms it into a new, useful representation. The great thing is that these kernels are not chosen beforehand, but that they are *learned* during the training of the network. That means that they are iteratively changed, so that they become more and more useful for the next layers in the neural network.

Let's have a look at what these CNNs can do.

## Segmentation with U-Net

In image processing, you often want to segment an image, that is, to divide the image up into different regions. These regions could be the lungs and heart in a chest X-ray, the vasculature in a retinal scan, or any other region of interest in an image. The U-Net is a type of CNN that was designed to do exactly that [1].

The first half of a U-Net consists of convolution layers, which look at the image features on a progressively larger scale. The second half then uses that information at different scales to produce the segmentation. A visualization of the network clearly shows how the U-Net got its name.



U-Nets are fully convolutional, meaning that there is no fully-connected layer at the end. An advantage of this is that the size of input images isn't fixed. That is especially useful in medical imaging, where datasets often consist of images taken by a mix of devices with varying output formats. Say you have a tool that can analyze lung X-rays for anomalies. You want to feed that tool with useful information: the area of the X-ray image in which you can see the lungs. In other words, you want to segment the image into two categories: *lungs* and *background*. This needs to be done automatically, using only the original X-ray image. The solution is to train a U-Net to do the image segmentation for you. The network learns by looking at examples, so you need a set of images and their segmentation.



Chest X-ray Image

(a)



Area to be segmented (U-Net)

(b)



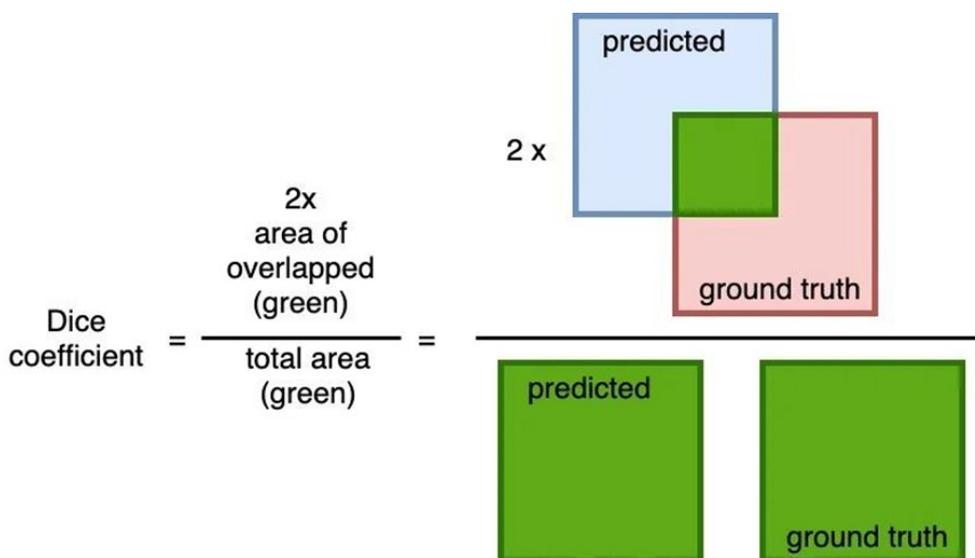
Segmented image

(c)

*Use a segmentation to mask the background, leaving just the area of interest. [3]*

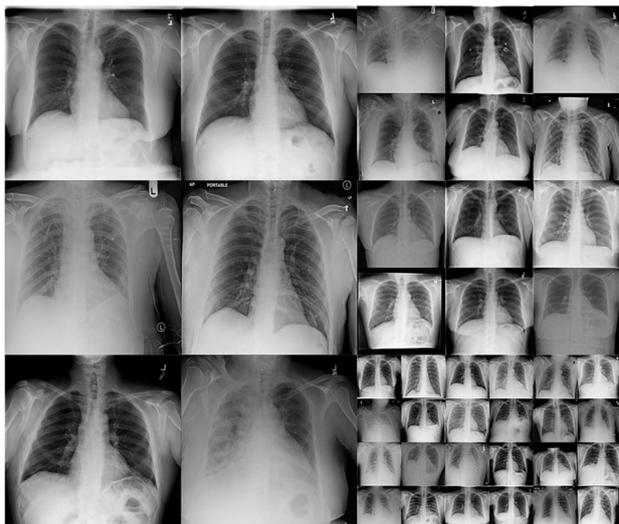
All you need to train a U-Net is a set of images and their segmentation, to serve as examples. These segmentations are often created manually by human labelers. The examples are then shown to the network time and time again, until it has learned to correctly segment the example images. It can then be used to separate the lungs from the background in X-rays it has never seen before.

To determine how well the model is performing, the model output must be compared with the original segmentation. An often-used metrics is the dice coefficient, which is based on the overlap between the two segmentations. If the Dice coefficient is close to 1 for a set of images that our model hasn't seen before, it produces the right segmentations and can be used! The model can now create segmentations for new images, that can then be overlaid with the image to select the relevant pixels.



## Creating synthetic data with GANs

Creating a large and varied dataset can be hard, especially in a domain where the data is as sensitive as in the medical field. On the other hand, these datasets are needed to properly train AI models. Wouldn't it be great if there was a way to increase the size of our dataset, without obtaining more sensitive data? Well, there is! Generative adversarial networks (GANs) are AI models that learn what a set of data looks like, and then output things that look like they belong in the same set. You might have already seen [this-person-does-not-exist](#), which generates photos of non-existent people, or you might have read Mathijs van Bree's [blog](#) on generating tabular data. Of course, these techniques can also be applied to medical imaging!



The image on the left shows a fully synthetic set of chest X-rays, generated using a GAN that was trained with real X-rays [4]. While they look good, the synthetic images are far from perfect. In 70% of cases, trained radiologists knew which image was fake when shown a pair of images (one real, one generated). The goal is to get that score to 50%, at which point the generated X-rays have become indistinguishable from real X-rays.

Synthetic data isn't just a way of creating a dataset without putting in the effort of data collection. It can potentially help in combatting bias in AI models. A big reason for bias in AI models is the underrepresentation of minority groups. Because AI model performance drops when there is not enough data, predictions about these groups will be wrong more often. GANs can be a way to increase the amount of data for a subgroup.

In medical imaging, there are many groups that can be over- or underrepresented, which impacts how fair an AI model will be. Cardiovascular disease is more common in men, but a diagnostic model should also work for women. Only few people have medical implants, but those should not cause a segmentation model to break down. GANs could become a tool for data scientists to ensure that their models don't just perform well on the majority group, but that they are also fair towards those outside of the majority.

### More for you

This blog barely touches the surface of AI in medical imaging, but hopefully it sparks some interest for the topic. If you want to know more about GANs, do check out [Mathijs van Bree's blog](#). If you are interested in other techniques for bias mitigation, this [blog by Almira Pillay](#) is a great place to start. If you are looking for inspiration and learning on a day-to-day basis, check out [WerkenBijSogeti](#). We are always looking for great minds to join us.

### References

- [1] [Ronneberger et al. \(2015\) - U-Net: Convolutional Networks for Biomedical Image Segmentation](#)
- [2] [Livne et al. \(2019\) - A U-Net Deep Learning Framework for High Performance Vessel Segmentation in Patients With Cerebrovascular Disease](#)
- [3] [Heo et al. \(2019\) - Deep Learning Algorithms with Demographic Information Help to Detect Tuberculosis in Chest Radiographs in Annual Workers' Health Examination Data](#)
- [4] [Segal et al. \(2021\) - Evaluating the Clinical Realism of Synthetic Chest X-Rays Generated Using Progressively Growing GANs](#)