COVID-19 CT Images Segmentation

Piotr Grabysz
Capstone Project Proposal

June 5, 2025

Abstract: Segment radiological findings on axial slices of lungs, by using data from a Kaggle competition.

1 Domain background

A computed tomography scan (CT scan), is a medical imaging technique used to obtain detailed internal images of the body. CT scanners use a rotating X-ray tube and a row of detectors placed in a gantry to measure X-ray attenuations by different tissues inside the body. The multiple X-ray measurements taken from different angles are then processed on a computer using tomographic reconstruction algorithms to produce tomographic (cross-sectional) images (virtual "slices") of a body [9].

The data used here are CT scans of the lungs of patients diagnosed with COVID-19. For such patients, computed tomography is a common stage in diagnosis. A radiologist is often asked to estimate the extent of damage relative to lung volume. It is a time-consuming procedure, because radiologist should look through all axial slices on CT and segment each of them [3].

2 Problem statement

In this project I want to design a model, which takes a CT scan as an input and outputs masks corresponding to two classes: *ground glass* and *consolidation*. According to Wikipedia, *ground glass* is

> an area of increased attenuation due to air displacement by fluid, airway collapse, fibrosis, or a neoplastic process. When a substance other than air fills an area of the lung it increases that area's density. On both x-ray and CT, this appears more grey or hazy as opposed to the normally dark-appearing lungs. Although it can sometimes be seen in normal lungs, common pathologic causes include infections, interstitial lung disease, and pulmonary edema [10].

whereas the *consolidation* means

a region of normally compressible lung tissue that has filled with liquid instead of air. The condition is marked by induration (swelling or hardening of normally soft tissue) of a normally aerated lung.

Consolidated tissue is more radio-opaque than normally aerated lung parenchyma, so that it is clearly demonstrable in radiography and on CT scans. Consolidation is often a middle-to-late stage feature/complication in pulmonary infections [11].

The process of creating binary masks is called image segmentation. Segmenting CT scans into *ground-glass* and *consolidation* is the goal of the Kaggle competition [3].

3 Dataset and inputs

The data comes from a Kaggle competition **COVID-19 CT Images Segmentation** [3]. At the time of writing, the competition is still active.

The dataset is built from two parts. The first part is a dataset of 100 axial CT images from more than 40 patients with COVID-19. The scans were labelled using a tool called Medseg [1]. The second part contains segmented 9 axial volumetric CTs from Radiopaedia [5]. It includes whole volumes and therefore both positive and negative slices (373 out of the total of 829 slices have been evaluated by a radiologist as positive and segmented). Sample scans and masks are presented in the Figure 1.

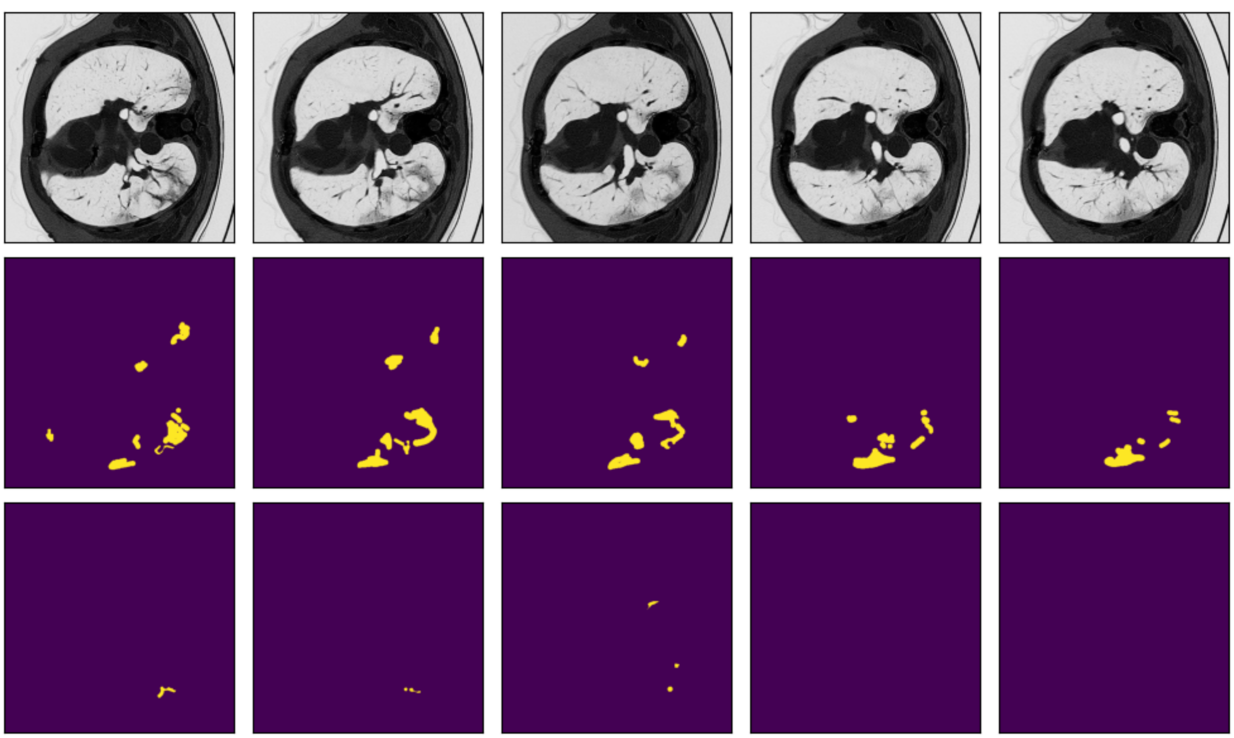

Figure 1: Sample images from Radiopaedia. The first row shows CT scans, second row: *ground-glass* mask and the third row shows *consolidation* mask.

Both parts consists of greyscale 512 × 512 images and masks with 4 channels: 0 – *ground glass*, 1 – *consolidations*, 2 – *lungs other*, 3 – *background*. Only the first 2 channels are relevant for this task. There are also 10 test images provided for the competition.

Images and masks are stored in arrays in numpy format. There are 5 files given in total:

- `images_medseg.npy`
- `masks_medseg.npy`
- `test_images_medseg.npy`
- `images_radiopedia.npy`
- `masks_radiopedia.npy`

4 Solution statement

In order to predict masks for computed tomography scans I will train a deep learning computer vision model. Because this is a problem of image segmentation, I will use U-Net architecture. This architecture was introduced in the paper *U-Net: Convolutional Networks for Biomedical Image Segmentation* [6] for segmentation of neuronal structures in electron microscopic stacks. The authors showed that such a network can be trained end-to-end from very few images. They used training strategy that relies on the strong use of data augmentation to use the available annotated samples more efficiently. It is also vital in this project, as I only have a few hundreds CT scans.

The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. See Figure 2.

5 Evaluation metrics

The metric is imposed by the Kaggle competition and it is *pixel-wise F1 score*. You can think about image segmentation as a binary classification for each pixel: it might either belong to a given class or not. F1 score is a metric used in classification and it is a harmonic mean of precision and recall:

$$F_1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2TP}{2TP + FP + FN}$$

where TP means True Positive, FP: False Positive and FN: False Negative.

In the context of image segmentation, the pixel-wise F1 score is sometimes called a Dice coefficient (or overlap index) and can be alternatively defined as

$$DICE = \frac{2|S_g \cap S_p|}{|S_g| + |S_p|}$$

where S_g is the ground truth image and S_p is the predicted image. The Dice coefficient is the most used metric in validating medical volume segmentations [7].

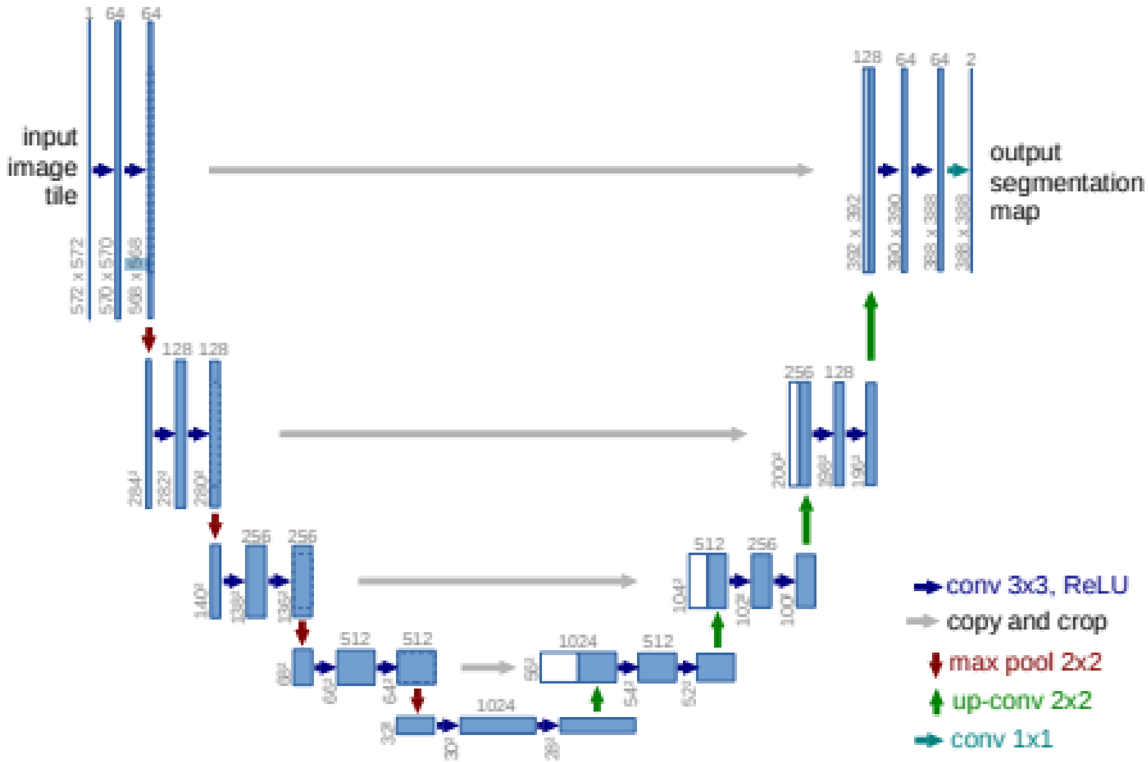

Figure 2: The original U-Net architecture [6]

Because the model needs to predict two classes (*ground glass* and *consolidation*), the score is averaged for both of them. The 10 test images will be used to calculate the Kaggle score and position myself in the Kaggle leaderboard. However, I will also divide the original training set into disconnected training and validation sets, so that I can assess performance of a model without Kaggle. This will be especially important for hyper-parameter search jobs.

6 Benchmark model

Because this is a Kaggle competition, then quite a natural benchmark is to compare my solution to the competition leaderboard. At the time of writing, the best model achieved a score of 0.73301. Of course, I cannot claim at this moment that I will beat the best model in the competition.

In the competition's "Code" section there is also a notebook showing a baseline model written in Keras [4]. It trains an out-of-the-box U-Net model. It achieves a score of 0.64561.

7 Project design

7.1 Model architecture

I plan to use the U-Net architecture, because it is a popular choice for CT scan segmentation, including lung segmentation in COVID-19 patients [2]. However, in contrast to the very first U-net described in [6], I will use a network with a pre-trained backbone. Such a solution is already implemented in a PyTorch framework called Segmentation Models (`https://github.com/qubvel-org/segmentation_models.pytorch`).

Using a pre-trained model should allow for faster training and possibly also better results, given a small amount of training data. I will start with the `efficientnet-b0` [8] as the backbone, because it is a small yet powerful network. If it will turn insufficient to solve to problem, I will try some bigger networks from the EfficientNet family.

Because the dataset is small, I will use a lot of data augmentation. Data augmentation is often used in computer vision when there is a limited amount of data and also the authors of the original U-Net paper claimed that data augmentation helped them to achieve strong results on medical data [6]. There are numerous options for augmenting images, like:

- Random rotation,
- Flipping (horizontal/vertical),
- Random cropping and scaling,
- Elastic deformation (used in the paper [6]),
- CLAHE (contrast-limited adaptive histogram equalization), used for grayscale images.

7.2 SageMaker implementation

The first step I will take is downloading the dataset from Kaggle, performing some EDA (Exploratory Data Analysis) and then uploading the competition's data to an S3 bucket. I will divide the original training data into separate training and validation sets. I will use validation set to compare performance of different models. I will sample validation images from the `image_medseg.npy`, because the Kaggle test data is taken from `medseg` as well, so I think it makes sense that the validation set is similar to the test set.

Then I will create a `.py` script containing the U-Net architecture and run a hyper-parameter search job with SageMaker AI script mode. The hyper-parameters which I plan to tune are *learning rate*, *batch size* and *encoder depth*, which is a parameter specific for U-Net implementation [1]. I will choose the best model based on highest pixel-wise F1 score. This metric, as well as other logs, will be saved by Amazon CloudWatch. Once I have the best set of hyper-parameters, I will train a model on the complete training set and then I will deploy it as an endpoint. I will process the test images and submit my predictions to Kaggle. An illustrative diagram of the whole process is shown on the Figure 3.

[1] You can take a look at U-net documentation here: `https://smp.readthedocs.io/en/latest/models.html#unet`

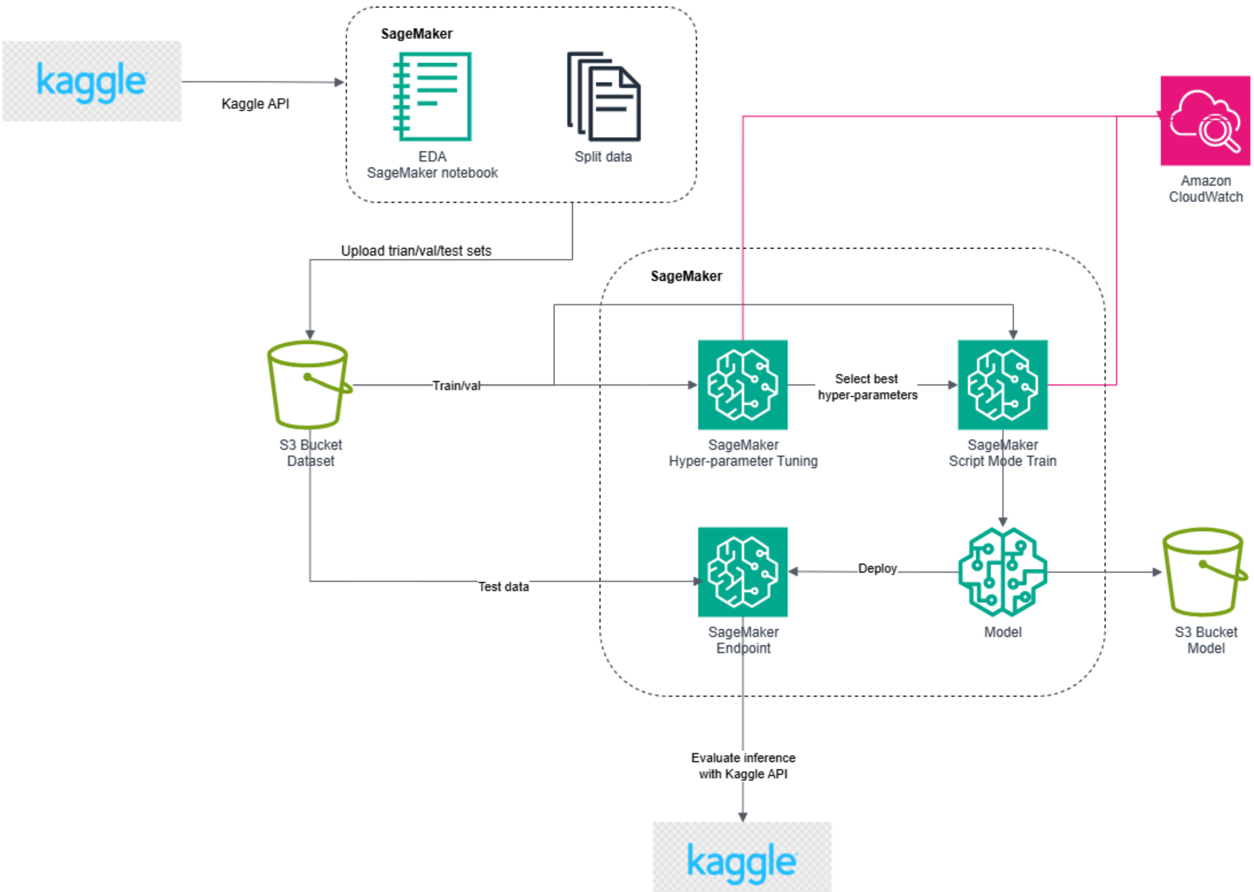

Figure 3: Diagram showing my AWS workflow

References

[1] https://medium.com/@hbjenssen/covid-19-radiology-data-collection-and-preparation-for-artificial-intelligence-4ecece97bb5b.

[2] Zhang F. "Application of machine learning in CT images and X-rays of COVID-19 pneumonia". In: *Medicine (Baltimore)* (2021). DOI: 10.1097/MD.0000000000026855.

[3] Igor.Slinko. *COVID-19 CT Images Segmentation.* https://kaggle.com/competitions/covid-segmentation. Kaggle. 2020.

[4] *keras baseline.* https://www.kaggle.com/code/yellowduck/keras-baseline. Kaggle.

[5] https://radiopaedia.org/articles/covid-19-4?lang=us.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation.* 2015. arXiv: 1505.04597 [cs.CV]. URL: https://arxiv.org/abs/1505.04597.

[7] Hanbury A Taha AA. "Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool." In: *BMC Med Imaging* (2015). DOI: 10.1186/s12880-015-0068-x.

[8] Mingxing Tan and Quoc V. Le. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.* 2020. arXiv: 1905.11946 [cs.LG]. URL: https://arxiv.org/abs/1905.11946.

[9] https://en.wikipedia.org/wiki/CT_scan.

[10] https://en.wikipedia.org/wiki/Ground-glass_opacity.

[11] https://en.wikipedia.org/wiki/Pulmonary_consolidation.