Semantic Segmentation of Colon Cancer Primaries

Julian Hugo
Hasso Plattner Institute
julian.hugo@student.hpi.de

Seulchan Hwang
University of Potsdam
seulchan.hwang@uni-potsdam.de

Daniel Paeschke
Hasso Plattner Institute
daniel.paeschke@student.hpi.de

Jonas Witt
Hasso Plattner Institute
jonas.witt@student.hpi.de

Abstract

Colon cancer is a very common cancer entity in developed countries with increasing incidence rates. Medical imaging is important for early colon cancer diagnosis, however interpretation of medical scans is a labor-intensive task. The application of deep learning approaches like Convolutional Neural Networks (CNNs) to semantically segment colon cancer tissue could substantially assist medical professionals in diagnosis and prognosis of this disease.

In this project, we compared two different model architectures (*U-Net* and *ResNet + U-Net*) and four different approaches to deal with class imbalance in the dataset. We used a labeled dataset of colon cancer primaries of computed tomography (CT) images provided by the *Medical Decathlon Challenge*. Our results showed superior performance of the pure *U-Net* model over *ResNet + U-Net*. Furthermore, we showed that data augmentation to alter the distribution of the input data has substantial influence on the performance of the *U-Net*. In alignment with our expectations, the *U-Net* model trained on an *oversampled* dataset yielded the best dice and binary performance metrics.

1 Introduction

Colon carcinoma is a very common cancer entity with especially high incidence and mortality rates in developed countries (e.g., in Germany: incidence 44.9 per 100,000 and mortality 21.7 per 100,000 in 2018) [1]. As the economic situation improves in many countries, the global health burden of colon cancer is expected to grow too [2]. Early diagnosis of colon cancer significantly increases the survival rates of patients [3]. Early colon carcinoma detection could be enabled by automated detection from CT scans, potentially using Convolutional Neural Networks (CNNs).

CNNs constitute a family of neural networks widely used in the field of computer vision (e.g., for classification and semantic segmentation). By integrating the spatial relationship of features and the notions of *translational invariance* and *locality* into a neural network, CNNs are especially suitable to process images where neighboring features (i.e., pixels, voxels) tend to be closely related [4]. The goal of semantic segmentation is to predict class labels for every single pixel of an image and can be seen as an extension of simple image classification [5].

Endocer-decoder segmentation networks (e.g., *SegNet*) were introduced in 2015 [6] and showed significantly improved segmentation performance compared to the previously applied *fully convolutional networks* (FCN) [7, 8]. They incorporate a decoder step which upsamples low resolution feature maps to segmentation masks [8]. The *U-Net* architecture implements *skip connections* between decoding and encoding steps [9].

Semantic segmentation using CNNs is widely used in the domain of medical imaging and shows promising results in clinical validation studies [10, 11]. Specifically the *U-Net* architecture (partially extended by *ResNet* [12] architectures) has been implemented to detect lung cancer [13], brain tumours [14] or liver cancer [15]. Current state-of-the-art medical image segmentation focuses on developing neural networks based on the *U-Net* architecture with various extensions [8].

The development of CNN architectures for successful medical image segmentation requires vast amounts of well annotated datasets and the resulting algorithms are often only applicable to specific segmentation problems [16]. The *Medical Decathlon Challenge* provided researchers with an open-source dataset collection containing ten medical imaging datasets of different human organs and cancer entities [16]. Numerous researcher teams participated in the challenge [17–19].

The overall aim of the *Medical Decathlon Challenge* was to build image segmentation models that generalize well — even for image classes and modalities which were previously unseen in model training. The evaluation was split into two phases: (1) The developed models were tested on a hold-out set of the provided datasets, (2) the models were tested on three, previously unseen datasets (Colon, spleen and hepatic vessels) [16].

In this project, we used the colon cancer dataset of the *Medical Decathlon Challenge* to predict segmentation masks of colon cancer primaries. We compared the performance of the established *U-Net* architecture [9] with a *U-Net* extended with a 34-layer residual network (ResNet-34) [12]. Furthermore, we investigated different approaches to overcome class imbalance problems.

Section 2 describes our approach to the segmentation task, including the applied data preprocessing, model architecture and performance metrics. In section 3, we compare the performance metrics of our different experimental settings. Section 4 discusses our model approach and outlines possible improvements. Our findings are summarized in section 5.

2 Methods

2.1 Dataset

The dataset used in this project stems from the *Medical Decathlon Challenge*, which is an open-source medical imaging challenge that provides medical images from different human organs or cancer entities and imaging modalities [16].

In our project, we worked on the dataset containing abdominal CT scans with segmentation mask labels of colon cancer primaries. Information acquired by CT scanning is expressed in dimensionless Hounsfield units (HU), the standardized radiodensity of the scanned tissue. HU are scaled so that air has a radiodensity of -1000 HU and water of 0 HU. The HU scale ranges from -1024 to 3071 HU [20]. CT images are three dimensional datasets, that can be sliced in axial, sagittal or coronal plane to create two dimensional representations of the images.

The colon cancer dataset consists of 190 CT scans, subdivided into 126 training and 64 test examples. The *Medical Decathlon Challenge* publicly provides segmentation mask labels only for the training examples. Thus, in this project we only used the training examples to be able to compute performance metrics on our predictions. All the CT scans were performed at the Memorial Sloan Kettering Cancer Center (New York, NY, USA), where patients received the scan prior to surgical cancer resection [16].

The *Medical Decathlon Challenge* emphasizes that the CT scans were obtained with different scanners, contrast agents, acquisition parameters and slice thicknesses. They state that the challenge of this dataset is its *heterogeneous appearance*. The segmentation masks were created manually by expert radiologists [16]. Each data sample is provided in the Neuroimaging Informatics Technology Initiative (NIfTI) image file format.

2.2 Data Preprocessing

2.2.1 2D Image Generation

We converted of the 3D information contained in the NIfTI files to axial 2D images, referred to as *slices*. We created slices of the image and its corresponding segmentation mask and saved each slice

as a numPy array (datatype for image was *int16*, for labels *uint8*). In model training we consider every slice as an individual, independent training example. Simultaneously we created a JSON file (*data_index.json*) that indicates for every single slice whether the corresponding segmentation mask contains any area labeled as cancerous tissue and which subset (test or training) it belongs to. The conversion from 3D data to 2D slices was done on training and test images.

2.2.2 Dataset Splitting

We split the whole dataset so that 10 % of slices were contained in the test and 90 % in the training subset. To ensure that the distribution of slices with and without cancerous tissue is the same in test and training set, we implemented a method that draws random slices so that the ratio between the classes is the same as in the original data. The resulting distribution of the split dataset can be seen in Table 1. In the training phase the training dataset was split further into 90 % training data (to build the model) and 10 % validation data (to calculate the validation loss).

Table 1: Dataset splitting

Subset	Slices	Cancerous slices	Non-cancerous slices
Training/Validation	12138	1157	10981
Test	1348	128	1220
Total	13486	1285	12201

2.2.3 Addressing Class Imbalance

As reported later the number of axial CT slices labeled with cancer tissue are significantly less than slices not containing cancerous tissue. This under-representation of slices with cancerous tissue reflects the reality of an abdominal CT scan, as colon cancer primaries appear locally in a section of the colon. However, we posed the question of whether this class imbalance of cancerous and non-cancerous slices might influence the performance of our model.

To investigate the influence of class imbalance on the performance of semantic segmentation in our dataset, we implemented a custom pytorch dataset class, that allows to apply four approaches to adjust the proportion of cancerous slices in the training dataset composition:

- *only_cancer*: Only slices containing cancerous tissue are loaded as training data.
- *oversample*: Increases the number of training examples of the minority class (slices with cancer) to match the number of the majority class (slices without cancer)
- *undersample*: Reduces the number of training examples of the majority class (slices without cancer) to match the number of examples in the minority class (slices with cancer)
- *original*: Uses the original unbalanced dataset as training examples

The exact number of training slices used in each approach is shown in Table 2.

Table 2: Class imbalance approaches

Dataset composition	Total slices	Cancerous slices	Non-cancerous slices
only_cancer	1157	1157	0
undersample	2314	1157	1157
oversample	24276	12138	12138
original	12138	1157	10981

2.2.4 Geometric Data Augmentation

To improve the generalization performance of our models on unseen data we implemented geometric data augmentation [21]. Here, the geometry of the image and the corresponding label mask is altered

equivalently. In our custom dataset class, we included the following transformations on the training images before model building in this order:

1. Resize: Scale the original image to 256 * 256 px
2. Resize: The image width and height is increased by 44 px
3. Random crop: The resized images are randomly cropped to the original image size
4. Horizontal and vertical flipping: Both transformations are independently performed randomly on the images

2.2.5 Data Normalization

To ensure that weight convergence is not impaired and to reduce the risk of optimizing for local minima, we applied a z-transformation of each image in the dataloader class [22]. Considering the standardized HU scale also when comparing images from different scanners, we used the overall training data mean and a respective average of batch standard deviations.

2.3 Model Architecture and Training

We decided to use a specific deep CNN, which — in alignment with its shape — is referred to as *U-Net* (see Fig. 3). It is an *encoder-decoder segmentation network* that is designed for the segmentation of biomedical images in an end-to-end setting [9].

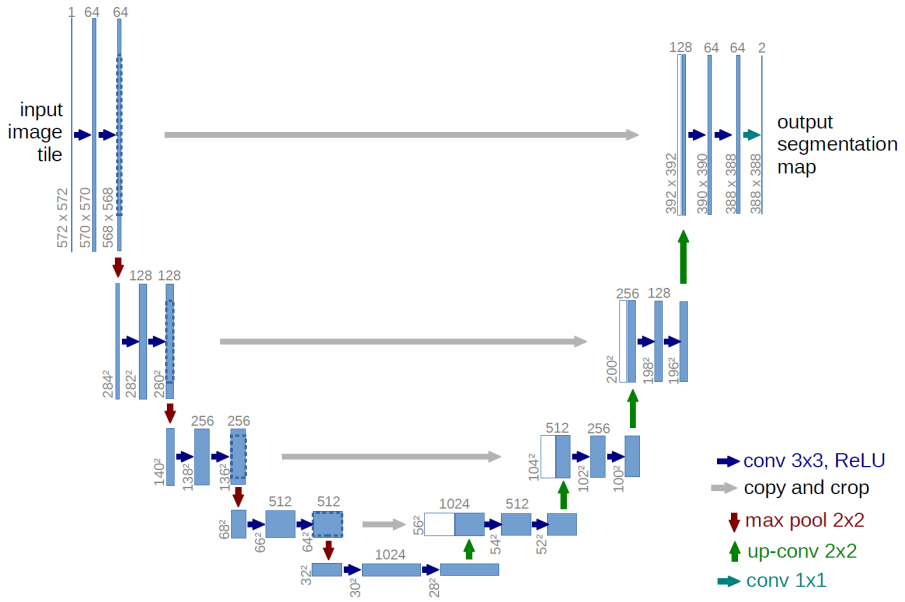

Figure 1: Original U-Net architecture, taken from [9]

In order to reconstruct a segmentation image, the U-Net consists of two sections. In the contraction path (encoder) the input images pass through a number of 3x3 convolutional layers (followed by the nonlinear activation function ReLU) and 2x2 max-pooling layers. The following expansion section (decoder) utilizes several respective up-convolutional layers in concatenation with the corresponding high-resolution features of the contraction section (see Fig. 3). These *skip connections* enable an accurate upsampling, eventually leading to the output of a high-resolution segmentation map. This, in our case, binary mask contains regions predicted to be colon cancer as 1 and background as 0.

One solution to overcome training difficulty problems of deeper neural networks (e.g. vanishing / exploding gradient) is referred to as *deep residual learning* (*ResNet*) [12]. The combination of a pretrained *ResNet* with the *U-Net*, where the *ResNet* constitutes the contraction path in the U-Net

architecture, has been shown to increase performance of the overall model [23]. We decided to compare the performance between *U-Net* and *ResNet + U-Net* on our data.

The pure *U-Net* model is trained in a single training phase. The *ResNet + U-Net* however is trained in two consecutive phases: (1) The pretrained weights of the *ResNet* are freezed, so that only the weights of the *U-Net* are updated during training. (2) *Fine tuning*: The pretrained weights of *ResNet* are unfreezed. The weights of the *ResNet* are pretrained on the ImageNet dataset [24].

To reduce complexity in this project we decided to treat the axial CT scan slices of all patients as independent examples, calculating merely 2D-convolutions. Hence, the adjacency of slices is not represented in this model. Furthermore, due to limited computational resources, we modified our networks to take 256 * 256 px 2D images as input (similar to [25]).

Eventually, we trained two different architectures on four different training dataset compositions (see 2.2.3) resulting in an overall combination of eight training procedures (see 2.5).

Table 3: Hyperparameters

Hyperparameter	*U-Net training*	*Res-Net fine tuning*
Batch size	12	12
Initial learning rate	0.001	0.00001
Learning rate patience (epochs)	10	10
Learning rate reduction factor	0.1	0.1
Maximum epochs	200 / 150	50
Epoch patience	30	30
Loss function	BCE + DCS	BCE + DSC
Optimizer	SGD	SGD

The selection of hyperparameters in model training largely influences the resulting model and its performance. We decided to implement two semi-automated approaches to determine suitable settings for the *learning rate* and *number of epochs*. To dynamically determine an appropriate learning rate, we implemented the PyTorch learning rate scheduler *ReduceLROnPlateau* into our training loop [24]. In our implementation, the initial learning rate is set to 0.001 and reduced by factor 0.1 if the validation loss did not increase for the last 10 epochs (*LR patience*). We used the default setting for the threshold, which determines the significance of changes in the validation loss (default: $1e-4$). In the *Res-Net fine tuning* phase we used the same settings apart from an initial learning rate of $1e-5$.
We defined 200 epochs as maximum number of epochs and to prevent overfitting we implemented used *early stopping* [26].
Furthermore, we chose to use a batch size of 12 for training and validation set. As optimizer we used stochastic gradient descent (SGD) [27]. SGD was used in the original publication of the *U-Net* [9] and is well established for semantic image segmentation [15, 28].
Our selection of hyperparameters is summarized in Table 3.

2.3.1 Optimization Metric

Using merely binary cross entropy (CE) as a loss function poses a challenge to train highly-imbalanced datasets with a small number of positive examples. The overwhelming effect of *easy-negative* examples may impair the loss function's sufficiency to train a neural network with ultimately great capability to segment the image effectively [23]. Thus, including the dice score into the loss (see 2.4) has been shown to improve training outcome [29, 30]:

$$L_{total} = L_{dice} + L_{CE}$$

2.4 Performance Metrics

The Dice Score (Sørensen–Dice coefficient / F1 score) is a widely-applied approach to measure the performance of segmentation algorithm. We decided to use this method, because it was proposed in the *Medical Decathlon Challenge*.

With X and Y representing *prediction* and *ground truth*, respectively, the Dice Score (DSC) is defined as

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

Analogously to the proposal in the *Medical Decathlon Challenge*, we are only considering the image foreground, hence the cancer-labeled pixels. To simplify the neural network, we were merely calculating 2D-convolutions (see 2.3), which is in contrast to other approaches tackling the original *Medical Decathlon Challenge*. Therefore, our dataset contains mainly examples without cancer. Hence, when the neural network correctly predicts on one of these slices that there is no cancer, the denominator of the DSC formula would equal zero and result in an error. To prevent this division by zero for correctly predicted non-cancer slices, we had to smooth the data accordingly by adding $+1$ to the numerator and denominator.

Additionally, we calculated metrics to evaluate the predictions image-wise (not pixel based). We analyzed whether our models predict cancer on a given slide in any location and interpreted this as a binary prediction: cancer anywhere on the slice vs. cancer-free slice. Based on these predictions, we calculated the precision, recall, false positive rate (FPR) and false negative rate (FNR). These measures do not provide direct information on the goodness of the segmentation, but rather assess if the models are able to identify suspicious slices, which might be relevant in radiology practice. Furthermore, we calculated the proportion of prediction masks that had an overlap with the ground truth mask of all slices that were correctly predicted as containing cancerous tissue (Overlap).

2.5 List of Experiments

In our project we want to compare two different CNN architectures and four training dataset compositions. Therefore we conducted the following experiments:

1. Compare architecture approaches:
   - *U-Net*
   - *ResNet + U-Net*
2. Compare class imbalance approaches:
   - *only_cancer*
   - *oversample*
   - *undersample*
   - *original*

3 Results

First, we explored the characteristics of our dataset and analyzed the proportion of cancerous slices.

Next, conducted the experiments outlined in section 2.5 to compare two different *U-Net* based CNN model architectures and four approaches to overcome class-imbalance problems. Our main goal was to identify the best combination of model architecture and data augmentation approach quantified by the previously defined dice and binary performance metrics.

The metrics of our models are summarized in Tables 4 (dice metrics) and 5 (binary metrics).

3.1 Data Exploration

We scanned all the values of our dataset and compared the range, to verify that our data is represented in HU, which was the case. Effectively the data could be stored in an array with datatype *int16*. All 126 data examples in our dataset had the same dimensions in the axial plane (512 * 512 px), whilst every example had a different number of axial slices. The number of axial slices ranged from 37 to 729 per example. The example containing 729 slices can easily be identified as an outlier. The

respective file contains two different abdominal CT scans merged together and therefore has more slices. As we considered each slice as an independent training example in our model, this did not cause any problems. In addition, we analyzed the number of slices that contain labels for cancer tissue. The number of slices containing cancer tissue range from 3 to 35 per example.

In total our whole dataset consists of 13486 axial slices. Of these 12201 are labeled as cancer-free and 1285 are labeled as containing cancerous tissue. Approximately 9.5 % of slices contain cancer tissue, causing a high class imbalance between the two classes (cancer vs. no cancer).

Figure 2 shows the distribution of axial slices per training example on the left and the distribution of axial slices containing cancerous tissue per example in the right histogram. X- and y-axes are scaled differently in both histograms to show the data most clearly.

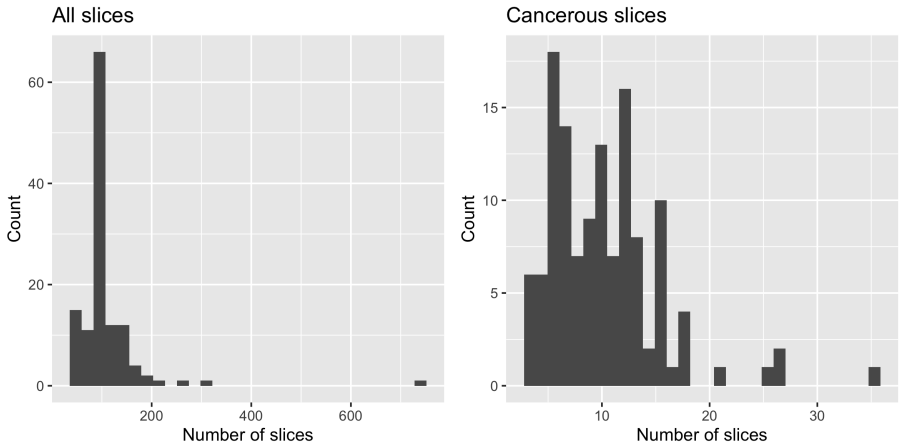

Figure 2: Histograms showing the distribution of slice numbers

3.2 Comparison of Architectures

We compared two different model architectures to predict semantic segmentation masks for colon cancer: *U-Net* and *ResNet + U-Net*.

Comparing the average dice score on the *only_cancer* and *oversample* datasets, the *U-Net* shows higher scores than the *ResNet + U-Net*. 0.686 vs. 0.342 on the *only_cancer* and 0.944 vs. 0.927 on the *oversample* dataset.

The average dice is the same for both models on the *original* dataset, as here both models predict all slices as cancer-free. Furthermore, *ResNet + U-Net* yielded a higher average dice on the *undersample* dataset: here the *ResNet + U-Net* model also always predicted cancer-free slices, resulting in a false negative ratio of 1.000.

As the general dice score performance of the *U-Net* is higher than the *ResNet + U-Net*, will only consider the comparison of the class imbalance approaches in the *U-Net* architecture in the next section.

3.3 Comparison of Class Imbalance Approaches

We compared four different training dataset compositions to find an optimal approach to address class imbalance. As outlined in the previous subsection *ResNet + U-Net* performed worse in terms of the average dice score, so for clarity we will focus on the performance metrics of the *U-Net* model on the different data augmentation approaches here.

The *original* dataset is highly imbalanced: only 9.5 % of slices contain cancerous tissue (see section 3.1). Using this as training set, the *U-Net* model always predicted segmentation masks without any cancer tissue, which leads to a dice score of 1.000 on non-cancer slices. Due to the high proportion of non-cancer slices in the test dataset, the average dice is high at 0.908. However, the false negative

rate of 1.000 reflects the one-sided predictions of the model. Precision, recall and overlap cannot be calculated as there are no slices predicted to contain cancerous tissue.

The *only_cancer* composition only contains cancerous samples. The *U-Net* models yielded similar dice scores for cancer (0.683) and non-cancer slices (0.686), average dice 0.686. Whilst recall is high (0.904), this model has a low precision (0.219) and a high false positive rate (0.329).

The *undersample* and *oversample* compositions are balanced datasets (50 % cancer and 50 % non-cancer slices). The *U-Net* architecture showed an average dice score of 0.897 (cancer slice dice of 0.365 and non-cancer slice dice of 0.951) on the *undersample* composition. On the *oversample* composition these three metrics were higher (average dice 0.944, cancer slice dice 0.508 and non-cancer slice dice 0.989) than on the *undersample* composition. Both these compositions yielded better dice metrics on cancer slices (compared to *original* composition) and on non-cancer slices (compared to *only_cancer*).

Also in terms of the binary metrics, the *U-Net* architecture trained with the *undersample* and *oversample* compositions yield better results, than when trained with *original* or *only_cancer*: E.g., precision was 0.520 for *undersample* and 0.849 for *oversample*. Training of the *U-Net* on the *oversample* composition yielded better precision, recall, FPR, FNR and overlap metrics than training on the *undersample* composition. Exact values of these metrics can be compared in Tables 4 and 5.

In summary, comparing both model architectures and all four training dataset compositions, the *U-Net* model trained on the *oversample* composition showed the best dice and binary performance metrics. Exemplary prediction results of this model are shown in the Appendix.

Table 4: Model Performance: Dice metrics

	DSC		
Model and composition	average	cancer slices	non-cancer slices
U-Net			
original	0.908	0.006	1.000
only_cancer	0.686	0.683	0.686
undersample	0.897	0.365	0.951
oversample	0.944	0.508	0.989
ResNet + U-Net			
original	0.908	0.006	1.000
only_cancer	0.341	0.444	0.330
undersample	0.908	0.006	1.000
oversample	0.927	0.427	0.978

Table 5: Model Performance: Binary and overlap metrics

Model and composition	Precision	Recall	FPR	FNR	Overlap
U-Net					
original	NaN	NaN	0.000	1.000	NaN
only_cancer	0.219	0.904	0.329	0.096	0.991
undersample	0.520	0.528	0.050	0.472	0.939
oversample	0.849	0.632	0.011	0.368	0.987
ResNet + U-Net					
original	NaN	NaN	0.000	1.000	NaN
only_cancer	0.119	0.920	0.693	0.080	0.870
undersample	NaN	NaN	0.000	1.000	NaN
oversample	0.726	0.616	0.024	0.384	0.974

4 Discussion

4.1 Performance of Our Model

In this project, we aimed to successfully segment colon carcinoma CT images provided by the *Medical Decathlon Challenge*. For our project, we merely used the respective training data, as the official test data labels were not available. The leading team *nnU-Net* achieved a dice score of 0.56 for colon cancer segmentation in the 2018 challenge[1] using a simple U-Net. We outperformed that metric substantially (*U-Net oversample average Dice:* 0.944). This substantial difference in performance metrics can be explained by the fact that the colon dataset was part of the second evaluation phase of the *Medical Decathlon Challenge* (see section 1). Contrary to the other teams we specifically trained for the colon cancer dataset and did not have to focus on entity generalization. On the other hand however, this was our very first deep learning project and we for sure could have integrated many potential improvements to our model (see 4.2).

In our experiments, the *U-Net* outperformed the combined *ResNet + U-Net* model. This is in alignment with the research team that overall performed best on the *Medical Decathlon Challenge* as they also used a simple *U-Net* architecture [30]. One promising alteration of the model architecture could be an *Attention U-Net*, which implements *attention gates* [31].

Previously, it has been shown that class imbalance can significantly influence the generalization performance of MLPs [32]. Common approaches to tackle class imbalance are *oversampling* and *undersampling*. In CNNs for classification tasks, *oversampling* has been found to be most effective approach [33]. In this project we showed, that also in semantic image segmentation the *oversampling* approach improves performance significantly.

As expected, the distribution of cancer and no-cancer slices in the training data is also reflected in the performance on the test data, i.e. *U-Net only_cancer* predicts cancer also on many non-cancer slices (high false positive rate) completely opposed to *U-Net original* that just predicted no cancer for all pixels (high false negative rate). Interestingly, *U-Net undersample*, having been trained with considerably less slices than *U-Net original*, while increasing the cancer slice proportion to 50%, results in almost the same average dice score as *U-Net original* (0.908 and 0.897, respectively).
Overall, the amount of training data used to build the models showed to be especially important: For *U-Net* and *ResNet + U-Net*, regardless of the distribution, the bigger our training dataset, the better it performed: *oversample* $>$ *original* $>$ *undersample* $>$ *only_cancer* (with the exception of *ResNet + U-Net original and undersample* that overfitted equally on non-tumor slices).

Overall, the heterogeneous appearance of the dataset (as emphasized by the *Medical Decathlon Challenge*) in terms of image properties has to be emphasized again. This taken into perspective, our models performed considerably well.

4.2 Possible Improvements

Several adjustments to our model approach could be made, that would presumably increase the model performance. The interpretation of medical tomography images is largely facilitated by taking adjacent image slices into account. Integrating the information of neighboring pixels not only in the axial plane, would result in more accurate encodings and is the common approach for our dataset [17–19].

Implementing *k-fold cross validation* into our training algorithm, would most likely help optimize our model hyperparameters. In the literature various research papers which applied *cross validation* for CNN training can be found [34, 35].

The selection of hyperparameters in model training largely influences the resulting model and its performance. Often hyperparameters are searched manually, however also partially automated approaches such as *grid search* exist [36, 37]. In this project we implemented to semi-automated approaches to determine suitable settings for the *learning rate* and *number of epochs*. However, more extensive *grid search* might yield optimal hyperparameter settings.

[1] http://medicaldecathlon.com/results.html

There is an ongoing debate on which loss functions enable the most effective and efficient optimization. Typically, several approaches are tried out sequentially to be compared later on. For this project we decided for an unweighted combination of dice and cross-entropy loss [30] (see 2.3.1). Still, other loss functions might be even more suitable. If this project would have been provided with more time and computational resources, it would be interesting to compare the performance of different loss functions like only-dice loss and some weighted combinations of dice and binary cross entropy.

During the project we discussed several options how to deal with the distribution of the data between the test, validation and training part. As we divided the data only once and did not use *k-fold cross-validation*, we decided to ensure equal distribution of cancerous and cancer-free slices between these subsets. In training a neural network, it is intrinsically assumed that the test data has on average the same distribution as your training data. Still, it can be argued to impair the test data. In our case it would be more realistic to test on whole patients than on equally-distributed slices.

We augmented our training data in different ways and compared the respective influence on the performance. Nevertheless, we would have liked to try out even more augmentations. Besides, the training datasets *oversample*, *undersample* and *only_cancer* somehow violated our equal-distribution assumption formulated in 2.2.2. With more time and computational resources we would have compared more approaches.

For medical image segmentation the *encoder-decoder neural network* architecture seems to be the most promising approach as of today [8]. Recent research focuses on improving the model structure, e.g. by modifying the *skip connections* [38]. Another approach is to apply *Neural Architecture Search* to identify suitable neural network architectures for semantic image segmentation automatically [39].

Last but not least, for reasons of limited computational power, we decided to decrease the resolution of our input images to 25 % of the original. Obviously, this also decreases the information contained in each single image. Research suggests that resolution degradation results in lower model performance [40]. Thus, with more computational resources we would suggest to retrain our best-performing model *U-Net oversample* on fully resolved images.

4.3 Challenges in the Project

The computing power that Google Colab provides for free is limited in memory and GPU power. Even the more stable GPU guaranteed by the Google Colab Pro subscription, did not manage to perform model training on the *oversample* and *original* datasets. Sometimes running calculations were terminated because of *inactivity timeouts*. Using a very powerful personal computer[2], the model training still took very long (up to 24 hours for one model). Obviously in a more professional setting, more computational power would be the key to build an optimal model.

Computer science literature research was new to some of us and it proved to be substantially different to doing literature research on bio-medical topics. Many hints on how to implement solutions were discussed in forums or written in blog posts. Some papers were published but not peer-reviewed yet. Overall, we are aware that deep learning constitutes a cutting edge research topic and working on it sometimes just includes trial and error. Nevertheless, we always tried to cite peer-reviewed papers whenever possible.

4.4 Future Work

Even though deep learning models applied on a variety of tasks have made remarkable progress recently, it has to be emphasized that specifically medical applications require a high level of accuracy and reliability. No AI-based algorithm can yet meet these requirements entirely as practically needed and a number of challenges remain. Still, progress in AI research continues and some applications are certified and tested to enhance physician's performances.

In radiology, the size of a solid cancer, its local growth and distant metastasis are criteria which are assessed to perform staging, determine therapy options and predict prognosis. In recent years,

[2]GPU: GeForce RTX 2060 TU104, 6 GB, GDDR6, 192 bit, GPU Clock: 1365 MHz, Memory Clock: 1750 MHz

research to support clinical decision making in this area has received increasing attention [41, 42]. As artificial intelligence approaches become more powerful in clinical decision support, its overall potential to improve healthcare has to be proven by the scientific community and medical professionals, the economical realization of business leaders and last but not least accompanied by a broad discussion in society [43].

5 Conclusion

In this project, we compared different approaches to train neural networks to detect colon cancer in CT images and perform semantic image segmentation, accordingly. We applied our approaches to the *Medical Decathlon Challenge* dataset using the colon cancer training dataset only. We compared two different architectures (*U-Net* and *ResNet + U-Net*) as well as four different training dataset compositions to address class imbalance. Performance on was measured by the dice score and several binary metrics.

We showed that the *U-Net* achieved the best overall performance when trained on the *oversample* training data composition. Generally it can be emphasized that the more data we used as input, the better the model performed. We showed that training on data with high class-imbalance impaired performance significantly.

Finally, we suggested a number of future improvements to our model approach.

References

[1] J. Ferlay, M. Colombet, I. Soerjomataram, C. Mathers, D.M. Parkin, M. Piñeros, A. Znaor, and F. Bray. Estimating the global cancer incidence and mortality in 2018: Globocan sources and methods. *International Journal of Cancer*, 144(8):1941–1953, 2019.

[2] David Holmes. A disease of growth. *Nature*, 521(7551):S2–S3, May 2015.

[3] ACS Atlanta. American cancer society. cancer facts & figures, 2020, 2020.

[4] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. *Dive into Deep Learning*. 2020. `https://d2l.ai`.

[5] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.

[6] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[7] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):640–651, 2017.

[8] Saeid Asgari Taghanaki, Kumar Abhishek, Joseph Paul Cohen, Julien Cohen-Adad, and Ghassan Hamarneh. Deep semantic segmentation of natural and medical images: A review. *Artificial Intelligence Review*, pages 1–42, 2020.

[9] 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.

[10] Miguel Monteiro, Virginia FJ Newcombe, Francois Mathieu, Krishma Adatia, Konstantinos Kamnitsas, Enzo Ferrante, Tilak Das, Daniel Whitehouse, Daniel Rueckert, David K Menon, et al. Multiclass semantic segmentation and quantification of traumatic brain injury lesions on head ct using deep learning: an algorithm development and multicentre validation study. *The Lancet Digital Health*, 2020.

[11] Kao-Lang Liu, Tinghui Wu, Po-Ting Chen, Yuhsiang M Tsai, Holger Roth, Ming-Shiang Wu, Wei-Chih Liao, and Weichung Wang. Deep learning to distinguish pancreatic cancer tissue from non-cancerous pancreatic tissue: a retrospective study with cross-racial external validation. *The Lancet Digital Health*, 2(6):e303–e313, 2020.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Siddharth Bhatia, Yash Sinha, and Lavika Goel. Lung cancer detection: A deep learning approach. In *Soft Computing for Problem Solving*, pages 699–705. Springer, 2019.

[14] Wei Chen, Boqiang Liu, Suting Peng, Jiawei Sun, and Xu Qiao. S3d-unet: separable 3d u-net for brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 358–368. Springer, 2018.

[15] Xiaomeng Li, Hao Chen, Xiaojuan Qi, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. H-denseunet: hybrid densely connected unet for liver and tumor segmentation from ct volumes. *IEEE transactions on medical imaging*, 37(12):2663–2674, 2018.

[16] Amber L Simpson, Michela Antonelli, Spyridon Bakas, Michel Bilello, Keyvan Farahani, Bram Van Ginneken, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, et al. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *arXiv preprint arXiv:1902.09063*, 2019.

[17] Fabian Isensee, Jens Petersen, Andre Klein, David Zimmerer, Paul F Jaeger, Simon Kohl, Jakob Wasserthal, Gregor Koehler, Tobias Norajitra, Sebastian Wirkert, et al. nnu-net: Self-adapting framework for u-net-based medical image segmentation. *arXiv preprint arXiv:1809.10486*, 2018.

[18] Dong Yang, Holger Roth, Ziyue Xu, Fausto Milletari, Ling Zhang, and Daguang Xu. Searching learning strategy with reinforcement learning for 3d medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 3–11. Springer, 2019.

[19] Mathias Perslev, Erik Bjørnager Dam, Akshay Pai, and Christian Igel. One network to segment them all: A general, lightweight system for accurate 3d medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 30–38. Springer, 2019.

[20] Tami D DenOtter and Johanna Schubert. Hounsfield unit. In *StatPearls*. StatPearls Publishing, 2019.

[21] Geoff Nitschke and Luke Taylor. Improving deep learning with generic data augmentation. *2018 IEEE Symposium Series on Computational Intelligence*, (SSCI):1542–1547, 2018.

[22] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[23] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[25] Albert Chon, Niranjan Balachandar, and Peter Lu. Deep convolutional neural networks for lung cancer detection. *Stanford University*, 2017.

[26] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[28] Dmitry Lachinov, Evgeny Vasiliev, and Vadim Turlapov. Glioma segmentation with cascaded unet. In *International MICCAI Brainlesion Workshop*, pages 189–198. Springer, 2018.

[29] Mahendra Khened, Varghese Alex Kollerathu, and Ganapathy Krishnamurthi. Fully convolutional multi-scale residual densenets for cardiac segmentation and automated cardiac diagnosis using ensemble of classifiers. *Medical image analysis*, 51:21–45, 2019.

[30] Fabian Isensee, Jens Petersen, Andre Klein, David Zimmerer, Paul F Jaeger, Simon Kohl, Jakob Wasserthal, Gregor Koehler, Tobias Norajitra, Sebastian Wirkert, et al. nnu-net: Self-adapting framework for u-net-based medical image segmentation. In *Bildverarbeitung für die Medizin 2019*, pages 22–22. Springer, 2019.

[31] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

[32] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[33] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

[34] Youyi Song, Zhen Yu, Teng Zhou, Jeremy Yuen-Chun Teoh, Baiying Lei, Kup-Sze Choi, and Jing Qin. Cnn in ct image segmentation: Beyond loss function for exploiting ground truth images. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 325–328. IEEE, 2020.

[35] Hao Dong, Guang Yang, Fangde Liu, Yuanhan Mo, and Yike Guo. Automatic brain tumor detection and segmentation using u-net based fully convolutional networks. In *annual conference on medical image understanding and analysis*, pages 506–517. Springer, 2017.

[36] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, and Vincent Dubourg. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[37] Marc Claesen and Bart De Moor. Hyperparameter search in machine learning. *arXiv*, page 1502.02127v2, 2015.

[38] Saeid Asgari Taghanaki, Aicha Bentaieb, Anmol Sharma, S Kevin Zhou, Yefeng Zheng, Bogdan Georgescu, Puneet Sharma, Zhoubing Xu, Dorin Comaniciu, and Ghassan Hamarneh. Select, attend, and transfer: Light, learnable skip connections. In *International Workshop on Machine Learning in Medical Imaging*, pages 417–425. Springer, 2019.

[39] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 82–92, 2019.

[40] Suresh Prasad Kannojia and Gaurav Jaiswal. Effects of varying resolution on performance of cnn based image classification: An experimental study. *Int. J. Comput. Sci. Eng*, 6:451–456, 2018.

[41] Daniel Hoklai Chapman-Sung, Lubomir Hadjiiski, Dhanuj Gandikota, Heang-Ping Chan, Ravi Samala, Elaine M Caoili, Richard H Cohan, Alon Weizer, Ajjai Alva, and Chuan Zhou. Convolutional neural network-based decision support system for bladder cancer staging in ct urography: decision threshold estimation and validation. In *Medical Imaging 2020: Computer-Aided Diagnosis*, volume 11314, page 113141T. International Society for Optics and Photonics, 2020.

[42] David Bouget, Arve Jørgensen, Gabriel Kiss, Haakon Olav Leira, and Thomas Langø. Semantic segmentation and detection of mediastinal lymph nodes and anatomical structures in ct data for lung cancer staging. *International journal of computer assisted radiology and surgery*, 14(6):977–986, 2019.

[43] Eric Topol. *Deep medicine: how artificial intelligence can make healthcare human again*. Hachette UK, 2019.

Appendix

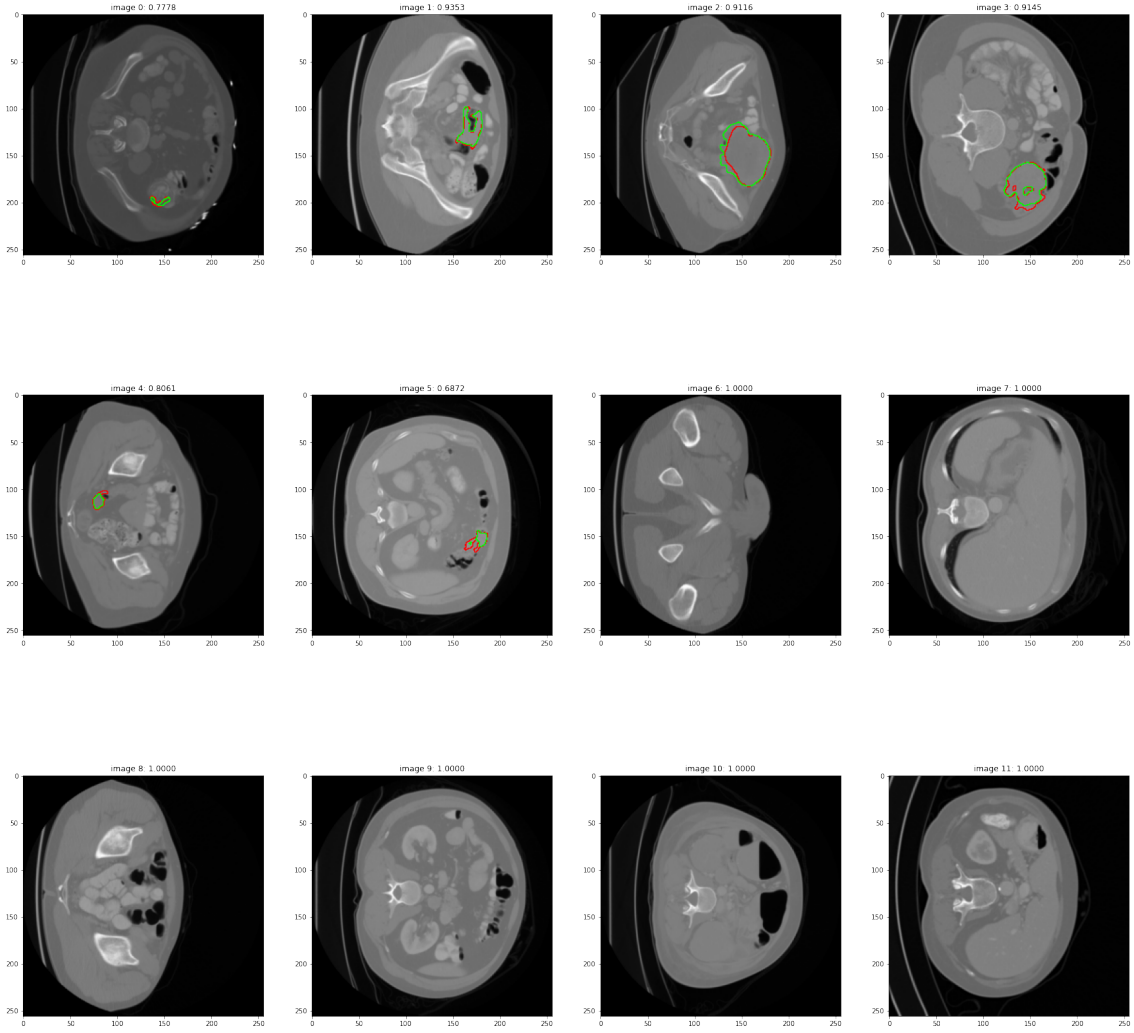

Figure 3: Exemplary radnomly selected segmentation mask predictions made by the *U-Net* model trained on *oversample* training data. The red lines show the ground truth and green lines the predicted borders of the segmentation masks. The average dice scores are shown above the respective image.