Prediction of brain tissues hemodynamics for stroke patients using computed tomography perfusion imaging and deep learning

Guillaume Barnier (005921957), Ettore Biondi (005957297), Greg Forbes (006190172), and Elizabeth Tong (n/a)
Stanford University
gbarnier@stanford.edu, ettore88@stanford.edu
gforbes@stanford.edu, etong@stanford.edu

Abstract

We use computed tomography perfusion data combined with a supervised deep learning algorithm to predict blood flow properties within the brain for ischemic stroke patients. We tackle the problem with a simple one-dimensional approach using three different model architectures and we show their potential at accurately predicting the important hemodynamic properties of the brain tissues.

1 Introduction

We focus on an advanced computed tomography (CT) technique known as CT perfusion (CTP) applied to patients suffering from ischemic strokes, in which blood flow through the artery providing oxygen to the brain becomes blocked. Within the brain, this leads to an irreversibly infarcted tissue (core) and a severely ischemic, but potentially salvageable, tissue (penumbra). However, there is a small time window (less than nine hours after the onset of the stroke) during which an accurate diagnostic must be made and therapy administered, such as reperfusion (Konstas et al. 2009).

In CTP imaging, a small bolus of iodinated contrast is injected intravenously into the patient. As it travels through the cerebral circulation, the contrast modifies the brain tissue density, which can be monitored with CT scans acquired every few seconds and provide valuable information about capillary-level hemodynamics (i.e., how easily the blood flows through tissues) within the brain.

To improve the efficiency diagnosing damages caused by strokes, we propose a supervised deep learning (DL) algorithm using 3D CTP images (our input) to predict a 3D image corresponding to the value of a hemodynamic parameter referred to as TMax at every position (voxel) of the brain. This parameter represents the delayed arrival of the contrast bolus and guides the radiologist to accurately predict and delineate the extent of the core and penumbra. We review limitations of existing solutions and describe our novel approach, present the dataset of interest and our preprocessing workflow, and apply our proposed algorithm on a data subset to show its potential at retrieving accurate TMax values for stroke patients. Finally, we analyze our results and discuss the next steps to improve predictions.

2 Proposed solution

2.1 Motivation

The conventional workflow to diagnose a patient follows a three-step process. First, CTP data are acquired by injecting an intravenous contrast bolus into the patient and scanning the head multiple times to follow the passage of the contrast through brain tissue (Ukmar et al. 2017). This 4D dataset is then fed to a software (e.g., RAPID) that returns 3D maps predicting TMax values at each location of the brain. Finally, a radiologist interprets these maps to delineate the core and penumbra.

CS230: Deep Learning, Winter 2021, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

This current approach presents three main limitations. First, the cost of such software may be high for certain hospitals. Second, parameter maps are computed using fluid-dynamics and (mathematical) deconvolution models with questionable assumptions. Finally, available software packages require fine time-sampling of CT images, exposing the patients to large radiation doses during acquisition.

To mitigate these drawbacks, we build an open-source DL framework in order to predict 3D maps of the TMax parameter value from 4D CTP data, as shown in **Figure 5** (pink box). Our goal is to reach an accuracy similar to RAPID without relying on any fluid-flow mathematical model.

While many authors have successfully applied DL methods on 3D CT data for image segmentation or classification (Barros et al. 2020; Park et al. 2019), our task here is of different nature because most of the useful fluid-dynamics information for predicting TMax is carried by the time axis of the data.

2.2 A one-dimensional approach

We choose to tackle this regression problem by casting it as a 1D time-series analysis. Similarly to RAPID, our algorithm processes the density curves (i.e., the variation of the density value at a given voxel as a function of time) of each voxel independently. Thus, one training example will be composed of one density curve of 80 time steps (our data) with its labelled TMax value computed by RAPID (and assumed as ground truth).

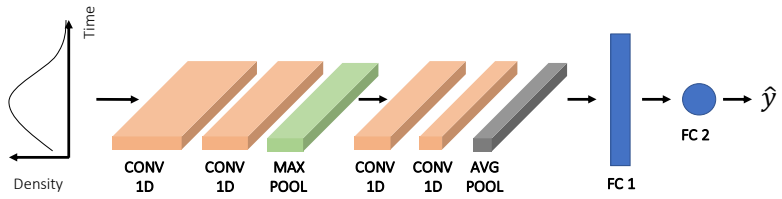
This simple approach allows us to leverage the amount of data as one patient can provide approximately 200,000 useful examples to train on, and data are available for approximately 400 patients. Moreover, 1D convolutions reduce the computational cost and enable us to iterate through the "idea/code/experiment" cycle more rapidly. Even though neglecting the spatial relations between samples may be detrimental, we first design and implement a simple and reasonable solution, assess its performance, and potentially address its limitations by incorporating more complex architectures. By following this strategy, we are able to gradually understand the challenges associated with these types of data, and get better insight on the possible ways to improve our results in the future.

2.3 Model architectures

We present and analyze the performance of four models based on two main types of architectures. We begin our analysis with two neural nets (referred as models 1 and 2) solely composed of fully-connected (FC) layers with various depths/widths as baseline references (**Figure 6**). We anticipated model 1 would not be large enough to fit rich training sets and model 2 might present overfitting.

For the second type of architecture, we follow the approach proposed by LLC 2021; Kiranyaz et al. 2021 where 1D time signal processing tasks are tackled with the use of 1D convolutional neural nets (1D CNN) to extract useful features from 1D time signals. Hence, models 3 and 4 are based on a sequence of 1D convolutional layers (1D CONV) followed by pooling layers. In both cases, the last two layers are fully connected. For all models, we employ ReLU activation functions (including the output layer). An example of model 4 is shown in (**Figure 1**).

Figure 1: Schematic of the 1D convolutional architecture (modified from LLC 2021). It is composed of two CONV 1D + POOL blocks followed two FC layers, for a total of 49,511 trainable parameters.



We tested three types of loss functions: (1) mean squared error (MSE), (2) mean absolute error (MAE), and (3) Huber loss function (HUBER). We chose to also examine (2) and (3) because we anticipated the presence of outliers in our TMax prediction in regions where our skull stripping procedure (detailed in the next section) was suboptimal. However, throughout our analysis, we achieved the best performance with conventional MSE.

2.4 Performance metrics

We propose two metrics to assess the algorithm's performance on our dataset. We compute a model accuracy value which is set to one for all predicted TMax values with a relative error smaller than a threshold value from the label, and zero otherwise. The threshold value was set to 20% based on medical insight. A potentially useful but more qualitative metric would be to have one panel of radiologists interpret the core/penumbra based on our predicted TMax, and compare the results with the interpretation of a second panel of radiologists using the RAPID output.

2.5 Hyperparameter search

We conducted an exhaustive and thorough hyperparameter search for each model. For this report, we only show models and parameter sets achieving the best performance. For more details, please refer to our github repository.

- **Conv. Layers:** For models 3 and 4, we varied the number of convolutional layers, the number of filters per layer, and the type of pooling layers (MAX and AVG).
- **Learning Rates:** For all models, we tested a wide range of learning rates as well as learning rate decay schedules: exponential decay, step decay, linear decay, and inverse time decay (along with each decay parameter).
- **Optimizers:** We tried three different optimizers: Adam, RMSProp and SGD with various batch sizes ranging from 64 to 1024.
- **Regularization:** We tried applying a weight decay (L_2-regularization) and dropout layers when we observed data overfitting (especially for model 2).

The hyperparameters which had most impact on our model performance were the depth of the neural net, the number of filters per convolutional block, and the learning rate value.

3 CTP dataset

3.1 Description

Our dataset was provided by the Stanford School of Medicine and contains CTP data for approximately 400 anonymous stroke patients, stored under the DICOM format. For each patient, it includes 55 3D CT images of the head acquired sequentially during the perfusion process with an acquisition sampling rate ranging from 0.5 s to 3 s. Each 3D image contains 10 axial slices of 512×512 pixels, representing the raw output of the scan. The vertical and horizontal samplings are 10 mm and 4 mm, respectively. In addition, the TMax 3D maps computed by RAPID are provided in a similar structure, but each axial slice is composed of 256×256 pixels, sampled at 8 mm.

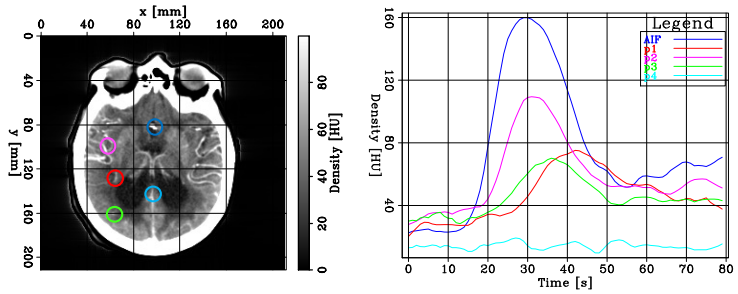
3.2 Pre-processing workflow

The raw amplitude data are converted to Hounsfield unit (Hounsfield 1973). By doing so, we easily identify image portions associated with the subject's head. We then rotate all the slices for a given time step to a reference orientation estimated on the central image slice. Moreover, as a given CT perfusion experiment can last a few minutes, the subject's motion is common to occur; therefore, we apply a co-registration to the time frames at a given depth with respect to a reference image slice. We perform this step by means of a skull mask created by thresholding (due to high-density of bone) and employ a phase cross-correlation method to identify relative shifts from the reference and other time frames. A moving spatial average of 2x2 pixels is applied to increase image coherency of each frame. Finally, we apply a time moving average filter of 6 s on each voxel to decrease experimental noise. To identify the portion of the CT scan slices related to the soft-brain tissues, we apply skull-stripping (Kalavathi and Prasath 2016).

Once pre-processing was complete, we reduced noise in the image for further enhancement imaging of the brain's perfused blood vessels (which were barely visible on the raw input) and remove the skull. The quality of our processing workflow is shown by examining density curves, the input for our algorithm. **Figure 2** shows five representative density curves (right) at different positions in the

brain (circled regions on the left) from the data. The dark blue curve on the right corresponds to the density recorded at the arterial pixel (the first pixel in the brain reached by the bolus).

Figure 2: Depth slice of a CT image extracted at $t = 30$ s (left). Colored circles indicate the extraction location of density curves as a function of time at different positions after pre-processing (right).



4 Training process and results analysis

The provided dataset is extremely large: after pre-processing, each of the 400 patients has approximately 200,000 useful training examples. We split the dataset into 380 patients for training, and 10 patients for both dev and test sets. Additionally, we examine and verify that the TMax distributions for the dev and tests sets are similar (by plotting their histograms). Since we did not have any prior experience processing these types of data and architectures, we conducted a three-phase training strategy where the amount of training data is gradually increased.

Phase 1: We train our models with a single axial slice (~23,000 examples) to ensure that each architecture is at least capable of fitting small datasets (without worrying about variance yet), which is a necessary (but not sufficient) condition. This phase was useful to gain better insight on the features of the predicted TMax maps. As anticipated, we observed (for all models) strong spatial granularity (i.e., noise) in the predicted maps from the lack of spatial information fed into the network.

Phase 2: We follow an analogous process and train all our models on a full patient head (~120,000 examples) where we subtracted one axial slice. We observed all models (except for model 1) managed to obtain a small bias, indicating that the architectures were complex enough to capture the dataset's main features. We evaluated the prediction accuracy on the subtracted slice and obtained high variance (as expected). However, predicted maps showed similar patterns as the ones obtained from RAPID.

Phase 3: We trained on 8 different patients' heads from the training set (~1,300,000 examples), and selected one patient from the dev set for hyperparameter tuning (~120,000 examples), and one patient for the test set. In this report, we did not include all patients in the dev/test sets due to time constraint, but we acknowledge that this should be done for future study.

After testing various hyperparameters during training, we found models 2, 3, and 4 (with their optimal hyperparameters) achieved similar accuracy and visual performance. **Figure 3** shows the MSE loss functions (left panel) and the accuracy of train and dev sets as a function of epoch during the training process for model 2. In both panels, the dev curves (red) are plotted as a reference, but the dev set is not used for training. The model reaches an accuracy of 46%, 43%, and 44% for the train/dev/test sets, respectively. As a comparison, the baseline model did not exceed 24% accuracy on the test set.

Figure 4 shows a sequence of three axial slices extracted from our TMax prediction at various depths from the test patient. The left column shows the prediction with model 2, the middle column is the RAPID output, and the right column shows the baseline prediction (model 1). As expected, model 1 performs poorly whereas models 2, 3 and 4 recovered the main trends present in the RAPID outputs. These results are encouraging considering that only 8 patients (out the 380 available) were employed for the training step. However, future work will include more data involve including more data.

Figure 3: Loss (left) and model accuracy (right) of train (blue) and dev (red) sets of model 2.

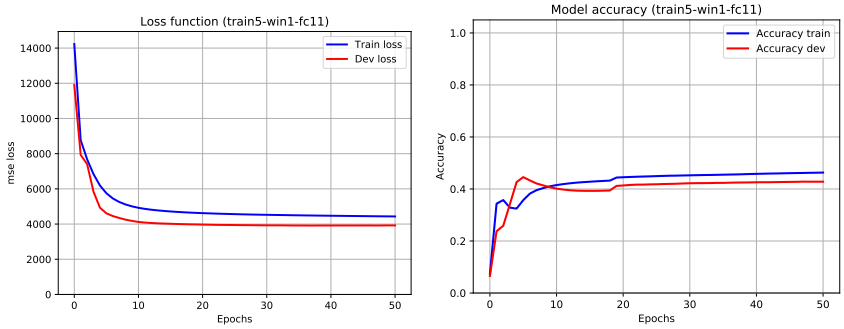
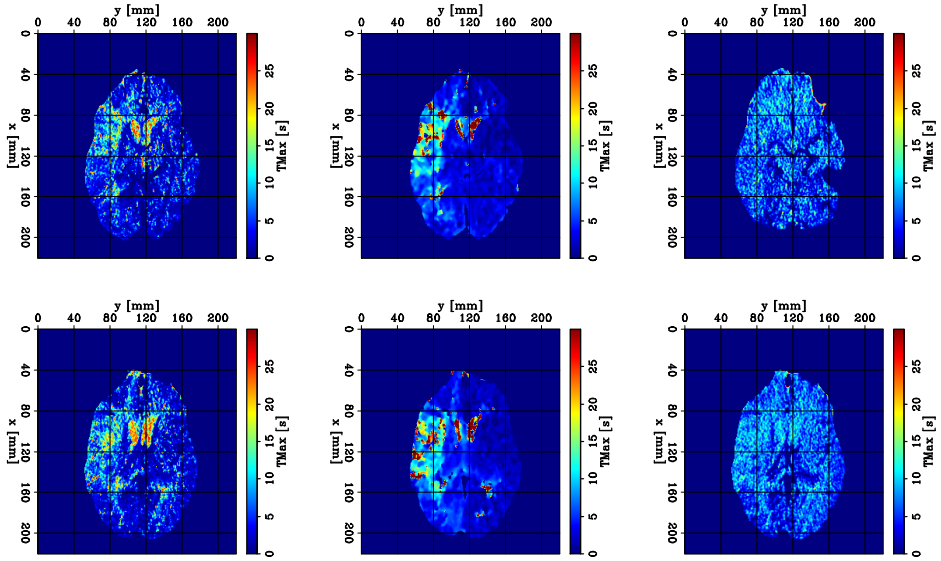


Figure 4: Depth slices of predicted TMax maps extracted from the test patient. Model 2 prediction (left), RAPID output (center), and baseline (model 1) prediction (right).



5 Conclusions, future work, and acknowledgements

We proposed a supervised DL algorithm to predict hemodynamic properties of brain tissues (TMax) for stroke patients using CTP data. We developed a simple 1D approach, which leveraged the amount of data and decreased the computational cost of the training process. The three models trained obtained similar performance and showed potential to recover the features from the RAPID outputs.

The development of a robust open-source DL algorithm as an alternative to existing commercial software can have an important medical impact. Tools to diagnose strokes will become accessible to more radiologists and may offer additional flexibility. For instance, such algorithms can potentially be trained to produce accurate results using less CT scans during the perfusion phase, thereby reducing the radiation received by the patient.

Finally, we see three main directions for future work. First, we will include more training data to improve robustness. Then, we would like to explore more-complex architectures (e.g., ConvLSTM) to enable embedding of spatial information between voxels and reduce granularity of predicted TMax maps. Finally, we wish to extend predictions to three other common hemodynamic parameters.

We thank the CS230 staff for the opportunity, Shahab Mousavi for his mentorship, and Pranav Rajpurkar, Eric Loreaux, Ashwin Agrawal, Bob Clapp, Fantine Huot and Stuart Farris for advice.

6 Appendices

6.1 Request for Project Confidentiality

The research presented in this paper is part of an ongoing research project. We request that the contents provided in this paper, as well as any part of the research provided in the other project submissions, be kept confidential and not shared amongst peers and classmates. Thank you.

6.2 Contributions

Guillaume Barnier: Guillaume Barnier was the main developer of the project architecture, ran models 1 - 4, and wrote the main drafts of the research submissions.

Ettore Biondi: Ettore Biondi developed the pre-processing workflow for the CTP dataset and edited the final drafts of the submission.

Gregory Forbes: Gregory Forbes helped with the development of the models, assisted with testing of hyperparameters, and edited the final drafts of the submission

Elizabeth Tong: Elizabeth Tong is the principal investigator for this project where she provided the students with the project strategy and gave her guidance throughout the project.

6.3 Additional figures

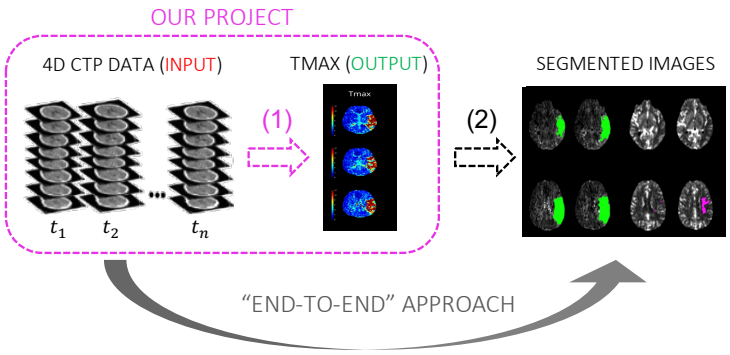


Figure 5: Schematic diagram representing how the challenging "end-to-end" approach for delineating the core/penumbra from 4D CTP data can be decomposed into two steps: (1) TMax computation from 4D CTP data (our project), and (2) image segmentation step (for future work).

Figure 6: Schematic diagrams representing the two architectures used as “baseline" for our analysis. The input is a 1D vector representing the density curve at a given voxel for all 80 time steps. The output $\hat{y}$ is our TMax prediction for that voxel. The first row (model 1) represents a neural net composed of one hidden layer of 100 hidden units (8,201 trainable parameters). The second row (model 2) is a neural net composed of 3 FC hidden layers of 500 hidden units followed by two hidden FC layers of 300 hidden units (782,401 trainable parameters).

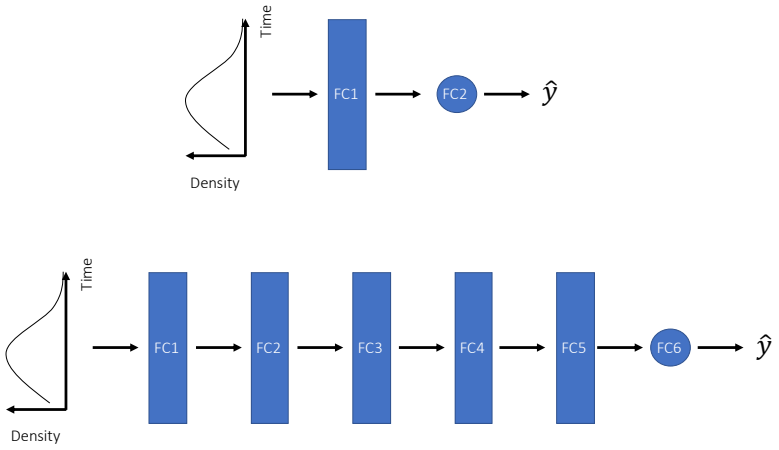


Figure 7: Schematic diagram representing the 1D convolutional architecture used for model 4 (modified from LLC 2021). It is composed of two CONV 1D + POOL blocks followed two FC layers, for a total of 49,511 trainable parameters. All CONV 1D layers in the first block are identical and composed of 30 filters of size 10. The MAXPOOL layer uses a window and stride of 3. The last two CONV 1D layers are composed of 50 filters of size 10. The average pool layer has a window length and stride set to 2.

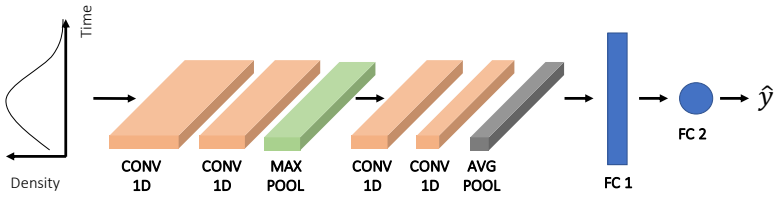


Figure 8: Schematic diagram representing the 1D convolutional architecture used for model 3 (modified from Kiranyaz et al. 2021). It is composed of three CONV 1D + MAX POOL blocks followed two FC layers, for a total of 33,337 trainable parameters. All CONV 1D layers are identical. They are composed of 24 filters of size 3, and a valid padding. All MAX POOL layers are identical with a window and stride set to 2.

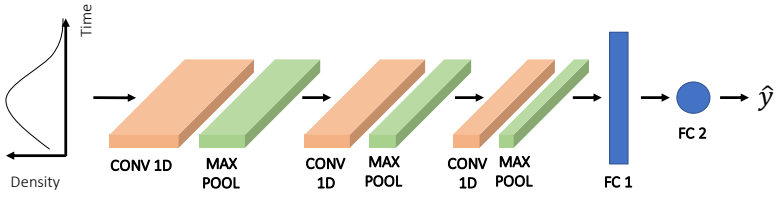


Figure 9: Axial slices extracted from the same CT image before (left panel) and after (right panel) our pre-processing workflow.

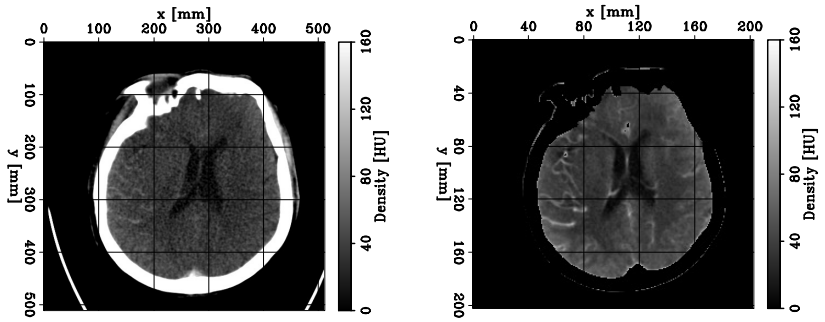


Figure 10: Depth slice of a CT image extracted at $t = 30$ s (top left). The colored circles indicate the extraction location of the different curves shown in the right panel. Density curves as a function of time extracted at different positions after pre-processing (top right). The dark blue curve is referred to as the arterial input function (AIF). The bottom panel shows the AIF before the pre-processing steps.

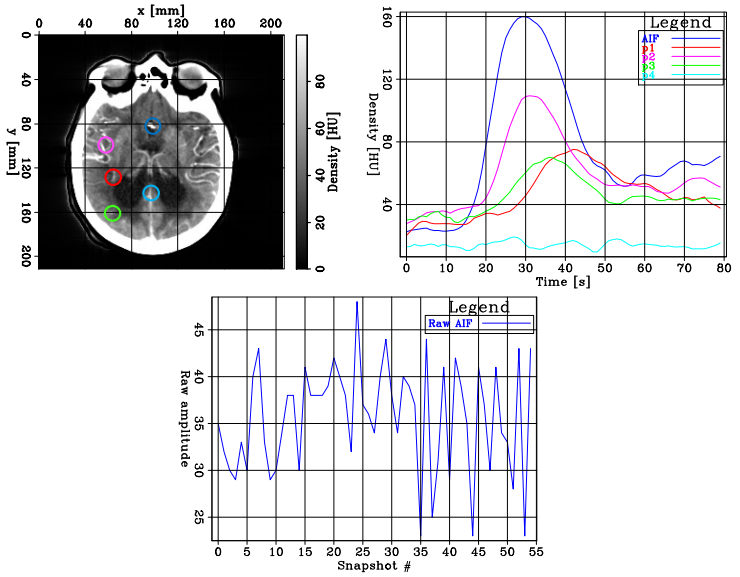


Figure 11: Convergence curves of the train (blue) and dev (red) sets as a function of epoch obtained for the training process of model 2. We use a learning rate $lr = 5 \times 10^{-6}$, a batch size of 512, a MSE loss function, and the stepping was done using the Adam optimizer. The computational time was approximately 10 min. The train and dev sets contained 1,300,000 samples and 120,000 samples, respectively. The left panel shows the loss functions and the right panel plot the model accuracy.

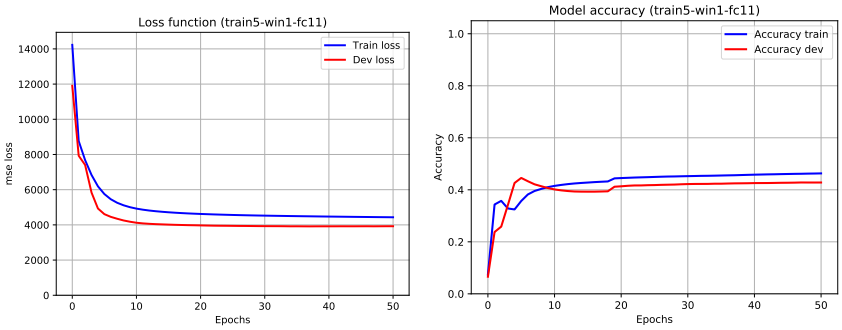


Figure 12: Axial slices of TMax maps extracted from the test patient at four depths. The left column displays the prediction of model 2, the middle column is the RAPID output, and the right column is the baseline (model 1) prediction. Each row corresponds to a fixed depth. The predictions were computed after training our models with only 8 out of the 380 patients available in our train set.