Semantic Segmentation on Dental X-ray Images

dlbs Mini-Challenge

Weiping Zhang

2023 HS Data Science

Fragestellung

Describe the initial situation and application to your question. To do this, explicitly formulate hypotheses, research questions or goals. The WHAT of your mini-challenge should be described here. Approximately 100-200 words.

Dentists spend a lot of time checking problems like tooth decay or gum disease. For proper diagnose, they need to be very careful to spot the small changes in the X-ray images that can show early signs of problems like small holes in the teeth or the beginning of gum disease, as well as get track of recovery process of some gum diseases, and make further treatment plan. Image segmentation models could potentially assist dentists for more accurate and efficient diagnosis, for example, segmentation and the teeth areas under the gum and outside of the gum would help dentists to track patients' oral conditions in long-term.

My focus is to create a binary image segmentation model specifically for dental X-ray images. Given the limited size of the dataset, the supervisor has suggested me to focus on binary segmentation. This means the model should distinguish all teeth as a single class at the pixel level within each X-ray image. The goal is to reach precise segmentation, teeth should be effectively differentiated from other elements in the image with high accuracy.

Datenlage

In this mini-challenge, I used the open access dental X-rays dataset with 32 classes (each tooth is a class) from humansintheloop. The dataset contains 598 x-ray images (3D), 598 json files with the objects information for each tooth, 598 masks generated by machine, and 598 masks generated by human (will not be used here). It includes both children and adult teeth X-ray images. The distribution of available teeth matches the reality, where wisdom teeth are the most common lost teeth, central incisors, canines are the least missing teeth. This suggests, the dataset is representative. For deep learning methods, this is a realtively small data size. After the discussion with the supervisors, I would decrease the segmentation class from 32 to 1. This means, it will be a binary segmentaion task.

Methoden/Vorgehen

TASK: Describe and discuss the choice of your solution approaches (baseline, DL architectures, evaluation) according to your question. Give an overview of the methods and parameters you use. That is, why did you do what you did? Approximately 300-500 words. The following sub-chapters should be answered directly in a notebook. Approximately 100-200 words are sufficient per [Kar19] discussion point.

Model: For this task, I have selected U-Net model, because it is particularly well-suited for tasks like medical image segmentation, due to its architecture could efficiently capture both context and localization information, and it can achieve good performance even with small datasets. For the baseline, I will use a simple architecture but self-trained parameters. For the second variant, I will use a pre-trained efficient-b4 model as encoder. This architecture is well applied on human chest X-ray image segmentation (Liu.W 2022 Automatic lung segmentation in chest X-ray images using improved U-Net). I use for both variants U-Net, because I would like to know if pre-trained model could really improve the model efficiency and accuracy on small dataset comparing to self-trained model.

Image augmentation: To reduce the overfitting risk due to small train data, I applied augmentation method to increase training size. The images in trainset will be transformed with random horizontal flip, vertical flip, rotation, adjust brightness and contrast.

Small dataloader: I also generated small dataset with only one batch for quickly testing the model's learning ability. If it works, I will train the model on full dataloader.

Adam optimizer: because it offers adaptive learning rates and can lead to quicker and more stable convergence. **Binary cross entropy loss with class weights:** In my images there are obviously more zero/black pixels than positive/white pixels. So, I added the class weights to the binary cross entropy loss to prevent bias towards the major class. Early stop: I know it is often used to prevent overfitting. However, here I used it simply to stop training process when the training loss is not improved within a certain number of epochs. As I would like to first train the model as good as possible even if it is overfitted. And afterwards I could apply regularization and dropout to improve the overfitting problem.

Save optimal models: Both models with the least validation loss and with the least train loss will be saved at the corresponding epoch. So that I have a choice to use the model with best performance on validation set or further apply regularization methods on the model with best train performance.

learning rate decay: If the training loss is not decreasing within a certain number of epochs, learning rate will be half reduced. Because in the early stages of training, a higher learning rate can help the model converge quickly. However, as training progresses, a high learning rate might cause the model to overshoot the minimum of the loss function. Reducing the learning rate over time could help to fine-tune the model's weights and avoid overshooting, leading to more precise convergence.

Wichtigste Resultate

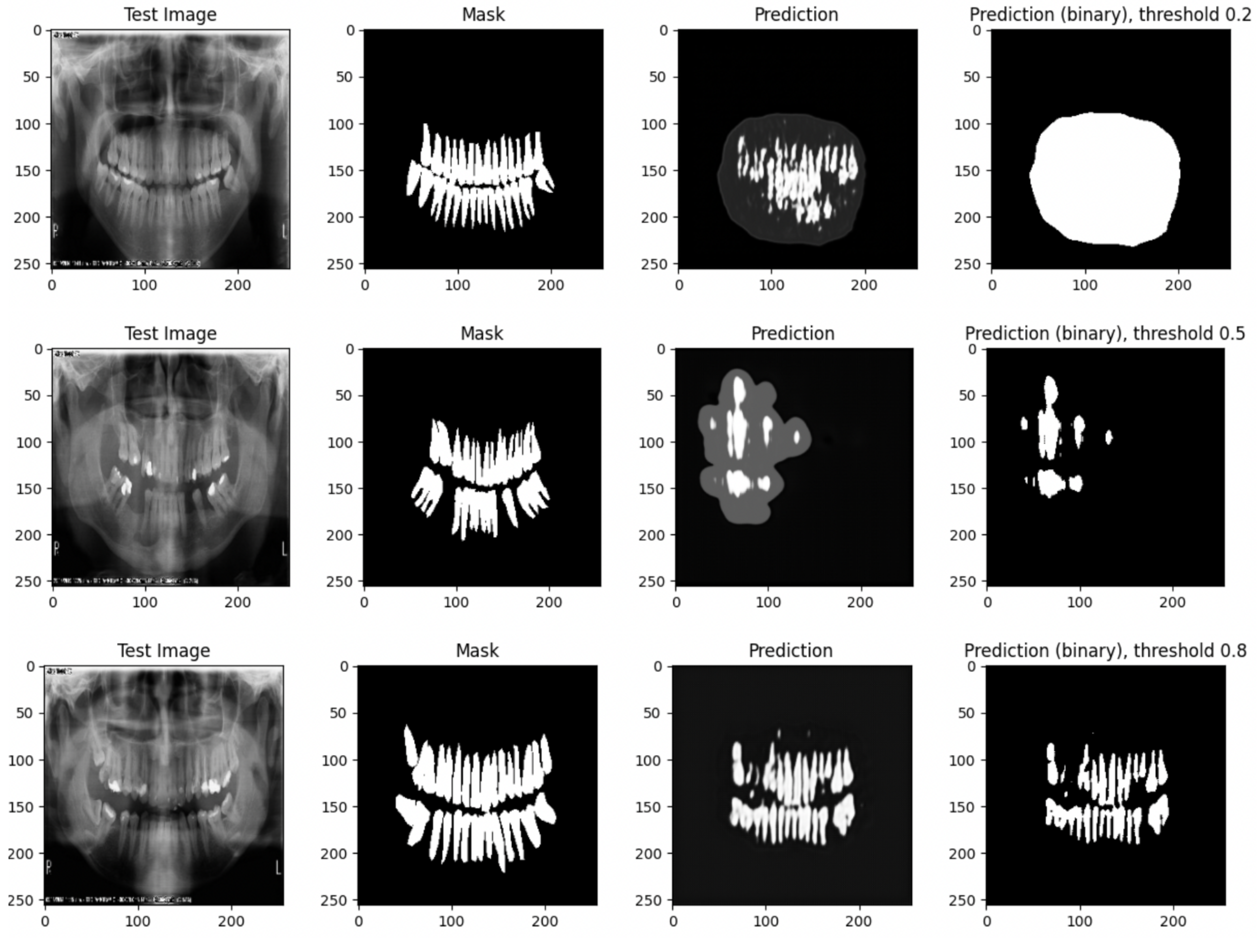
Test Image
Mask
Prediction
Prediction (binary), threshold 0.2
Test Image
Mask
Prediction
Prediction (binary), threshold 0.5
Test Image
Mask
Prediction
Prediction (binary), threshold 0.8
0
50
100
150
200
250
0
100
200

Figure 1. Tuning prediction threshold

In the figure 1, we could see predicted images are with pixel values between 0 and 1. A threshold is needed to convert the values to binary. All predicted pixel values that lower than the threshold will be considered as 0, all others as 1 (white). Here we could see that, with threshold 0.2, many background pixels are falsely converted to 1. With threshold 0.5, the gray pixels are converted to black. With threshold 0.8, it split the background pixels and positive pixels with the best.

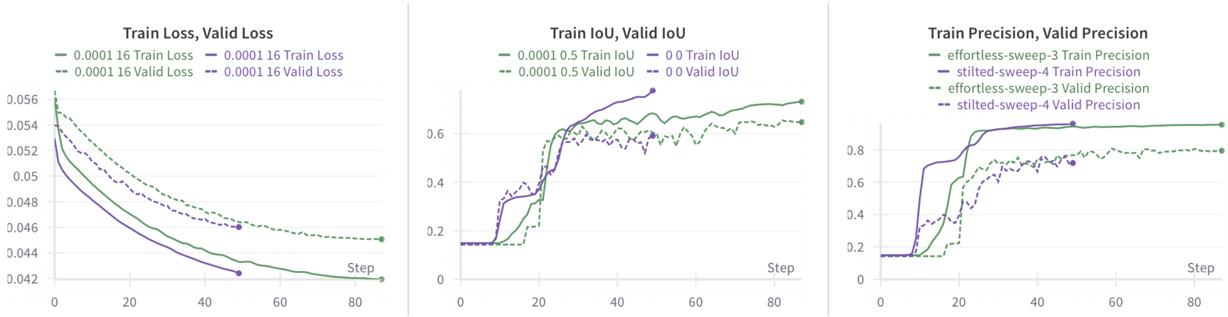

Figure 2. Improve model generalization with regularization and dropout (U-Net-efficient-b4 with 1 x augmentation)

To improve the generalization of my overfitted model (Fig. 2 in purple), I used L2 regularization which will add a penalty to the loss function and dropout which will drop a certain neurons to forces the model less sensitive to specific weights of neurons. With the grid searching of regularization strength of 0.01, 0.001, 0.0001 and dropout rate of 0, 0.2, 0.5, 0.8, finally I found with regularization strength of 0.0001 and dropout rate of 0.5, the model could largest improve the performance on the validation set. We could see that the green lines have better validation IoU and precision, and smaller gap between training and validation set. With the optimal regularizaiont strength and dropout rate, I will train the model with more epochs.

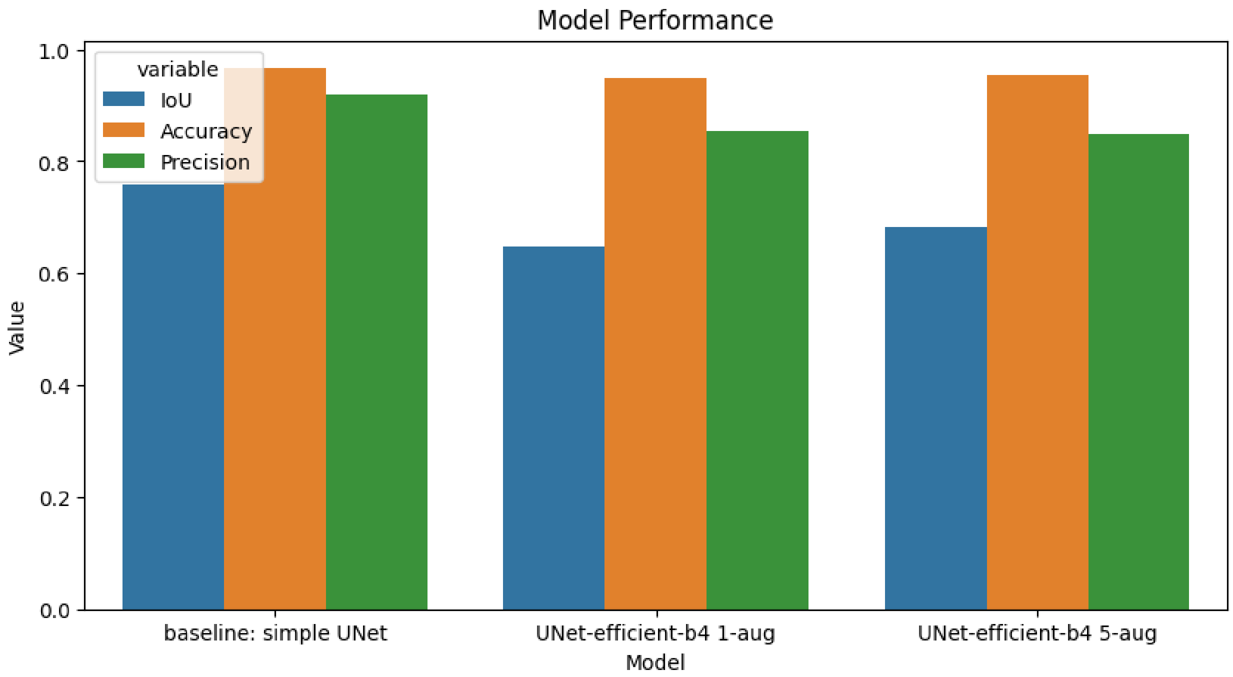

Figure 3. Compare the three models performance

All three models baseline model (Fig. 3) with 1 time augmentation, U-Net-efficient-b4 with 1 time augmentation and 5 time augmentation show similar high performance of accuracy and precision. The simple UNet model has better IoU than the other two models. This suggests, though the efficient-b4 encoder is trained based on a large set of data, training on specific dataset could generate better result. Comparing the middle and right models, we could see that applying 5 times augmentation where each train image will get 5 transformed images has improved the IoU performance.

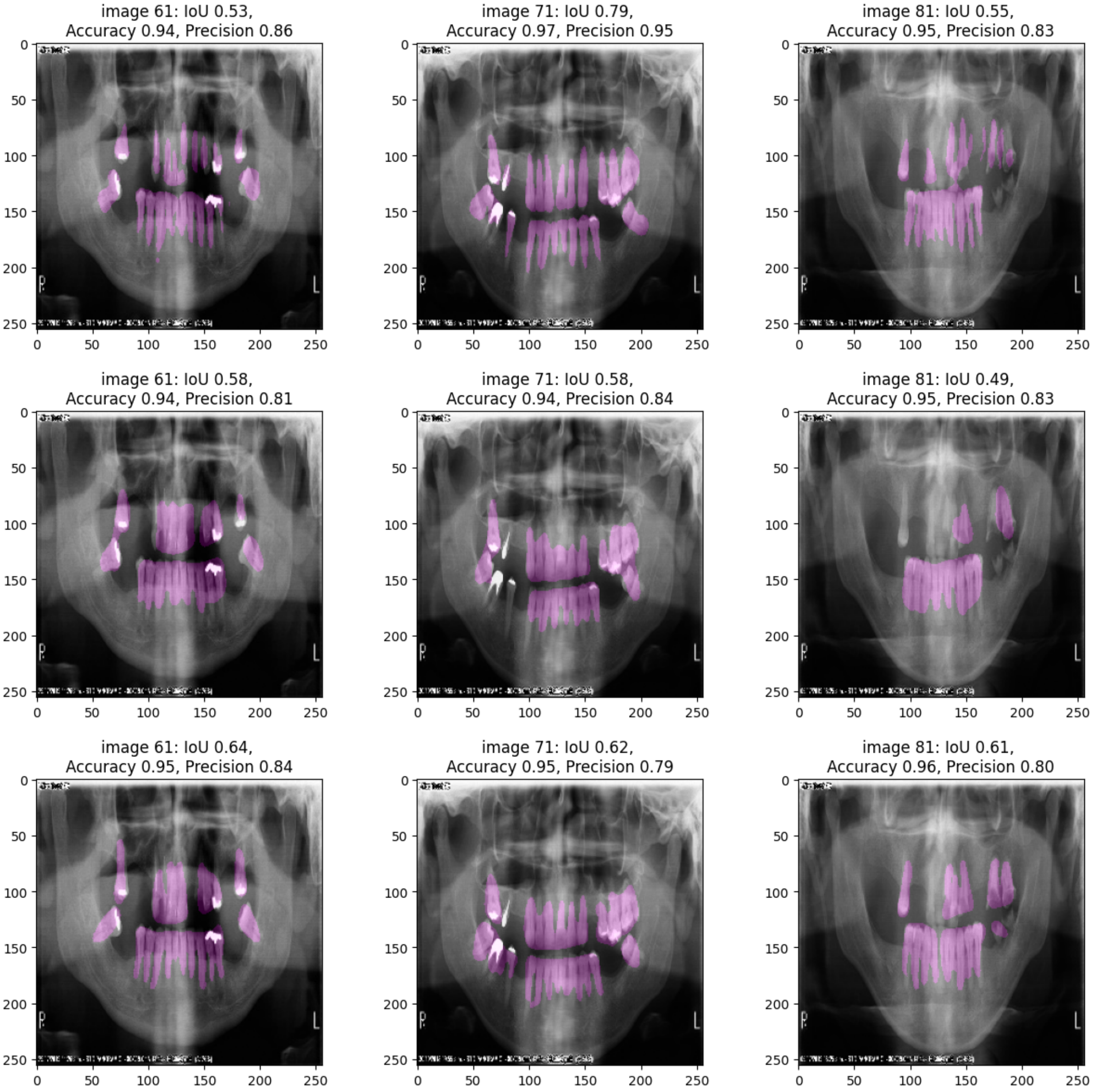
image 61: IoU 0.53,
Accuracy 0.94, Precision 0.86
image 71: IoU 0.79,
Accuracy 0.97, Precision 0.95
image 81: IoU 0.55,
Accuracy 0.95, Precision 0.83
image 61: IoU 0.58,
Accuracy 0.94, Precision 0.81
image 71: IoU 0.58,
Accuracy 0.94, Precision 0.84
image 81: IoU 0.49,
Accuracy 0.95, Precision 0.83
image 61: IoU 0.64,
Accuracy 0.95, Precision 0.84
image 71: IoU 0.62,
Accuracy 0.95, Precision 0.79
image 81: IoU 0.61,
Accuracy 0.96, Precision 0.80
0
50
100
150
200
250
R
L

Figure 4. Compare the prediction details of the three models: first row is baseline model, second row is U-Net-efficient-b4 with 1x augmentation, third row is U-Net-efficient-b4 with 5x augmentation.

From figure 4 we could see that, all three models could recognize all the teeth in the first image. In the second image, the U-Net-efficient-b4 with 1x augmentation has difficulty to distinguish the high brightness teeth, same as in the third image, two teeth were not recognized. The first row and third row could predict better on the small details, and the tooth with complex information or specific form, for instance, the two very bright teeth in second image. They could also detect the teeth crown and neck very well. But it seems all three models have difficulties to distinguish the deep roots from the background. This may due to the low contrast between these structures and the background.

Diskussion

All my models could predict the pixel classes with IoU larger than 0.64, precision larger than 0.95. The models could in general detect most of the teeth though with inaccurate area and boundaries.

My final optimal model is the baseline model, a simple U-Net, it could detect almost all of the teeth, especially the teeth crown and neck could be accurately predicted. In very rare cases, few teeth were not detected at all. However, the deep roots of teeth in general could not be accurately recognized, because the pixel values are too close to the background. It also falsely predicted some background spots as positive (teeth), e.g. at the bright non-teeth spots, and the space between two teeth. In reality, this may lead false diagnosis and treatment decision for the patients. Because tracking patients oral health requires high accurate segmentations even on small details.

My optimal model has IoU of 0.76 on unseen data, indicating that, on average, 0.76 of the positive pixel overlap in the predictions is correct. Precision of 0.92 suggests that my model is reliable in its positive predictions, with fewer false positives. An accuracy of 0.97 means that 0.97 of the pixels are corrected predicted, but in this case it is misleading because my data is imbalanced.

Model architecture: My second architecture uses a pre-trained encoder based on large amount of data. The advantages are it could extract features of an image more efficient and save computing time. However, my self trained model works much better than the pre-trained architecture. The reasons could be 1. the parameter of encoder block of the simple UNet is updated in training process. But in the pretrained-efficientnet-b4 UNet, I used the pretrained parameters for encoder block. This means, in this case, self trained encoder parameter is more suitable to my images than the pretrained parameters. 2. The UNet with pretrained model has more complex middle sub block and decoder subblock than baseline. A more complex model needs a larger training dataset to reach good performance.

Improvements:

Data size and diversity: In this challenge, I have apply image augmentation method to increase train data size. With 1 time (2 x train size) and 5 time (6 x train size) augmentation, the model performance has been improved. To further improve the model, it would be good to introduce more images from other data source for a larger data diversity.

Fine tuning of learning rate: With tuning learning rate and batch size I have find the suitable hyperparameter and used in configuration. To further improve the model performance, further fine tuning the learning rate could help model be even closer to the global minima.

Update parameters of pre-trained model: This means integrate the pre-trained encoder's architecture into the training process, but not its pre-trained parameters. The reason is the second model has indeed very good CNN architecture, which could capture the image features efficiently. By training this architecture from scratch on my specific dataset, the parameters will be updated during backpropagation to become more specific to my data. The model will be able to better capture informative and relevant features specific to the dental X-ray images.

Reflexion

My work could distinguish the teeth from other parts with a good accuracy. However, to implement this method in dentistry, depending on the specific goals, more classes must be segmented with very high accuracy, because it has to be able to track small changes accurately to assist the dentists doing the diagnosis more efficient. I found myself very motivated in the step of looking for a suitable project. By searching and reading, I realized implementation of deep learning image segmentation does not have to be in the field of auto-driving, big data fields. It could use small dataset to improve work efficiency in many fields. In this challenge, the model training part is running good. But I would prefer choosing another baseline model next time instead of two variants of U-Net.