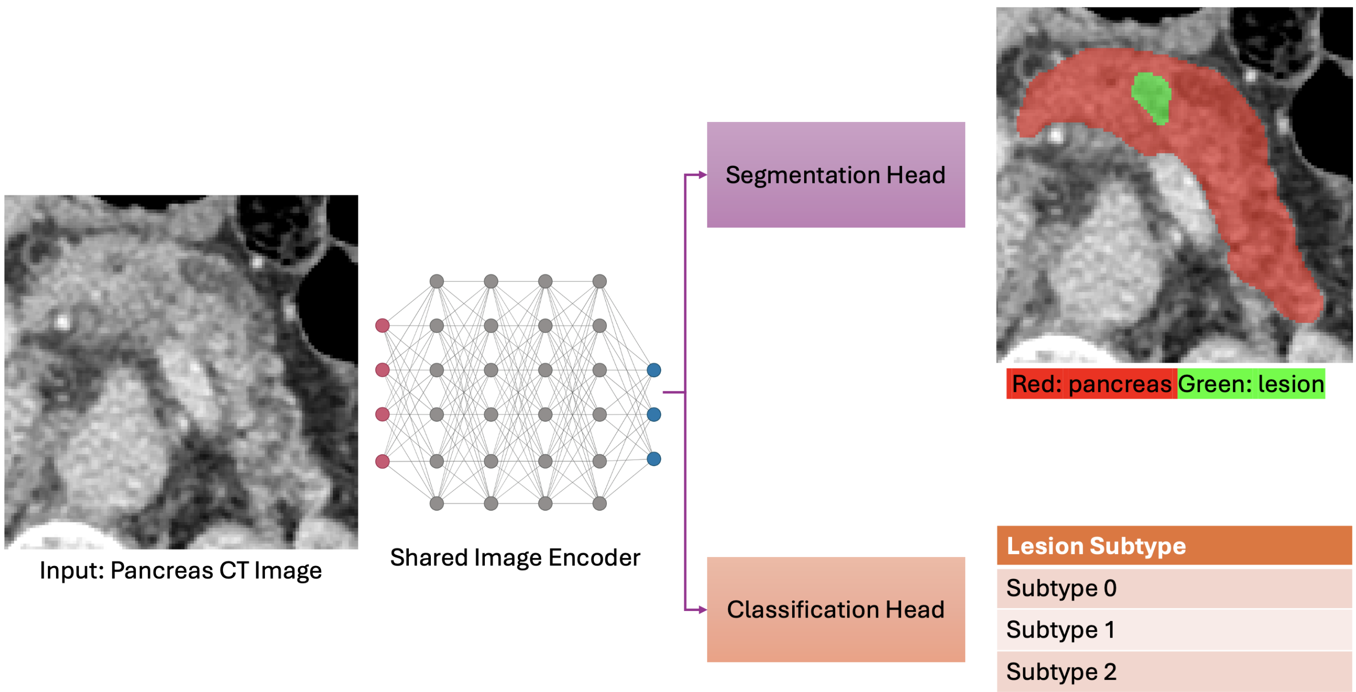
Deep Learning for Automatic Cancer Segmentation and Classification in 3D CT Scans

Task Introduction

This quiz is designed to test your understanding of deep learning techniques specifically applied to medical imaging, focusing on the segmentation and classification. The task covers various aspects of the deep learning fundamentals, including convolutional neural networks (CNNs), image preprocessing, data augmentation, model evaluation, and specific challenges associated with medical image analysis (heterogeneous appearances and resolutions).

We provide a de-identified pancreas CT dataset with segmentation annotations and lesion subtype labels. (three classes). The objective is to build a **multi-task deep learning model** for pancreas cancer segmentation (label 1: normal pancreas (red), label 2: pancreas lesion (green)) and classification.



The model should have a shared encoder to extract image features and two separated decoder head for segmentation and classification, respectively. Please develop the model based on the popular nnUNetv2 framework, which already provide state-of-the-art segmentation networks and out-of-the-box image preprocessing and model training pipelines. You only need to modify the architecture to implement a classification head. In addition, please implement at least one strategy to improve the default nnUNetv2 inference speed. You can find many ways to improve the efficiency from the FLARE22 and FLARE23 challenge winning solutions.

After finished the network design, you can train it on the provided training dataset followed by inferring on the validation and testing sets. The evaluation should use the suggested metrics in Metrics Reloaded.

Dataset

In order to enable tasks to be completed on publicly available free computing resources (e.g., T4 on Colab and K80GPUs on Kaggle), the original large 3D pancreas CT scans have been cropped to small region of interests (ROIs).

Dataset Splits

Split	Subtype 0	Subtype 1	Subtype 2
Train	62	106	84
Validation	9	15	12

Folder Structure

```
data
├── train
│   ├── subtype0
│   │   ├── quiz_0_041.nii.gz # mask (0-background; 1-pancreas; 2-lesion)
│   │   ├── quiz_0_041_0000.nii.gz # image
│   │   ├── ...
│   ├── subtype1
│   └── subtype2
└── validation
│   └── subtype0
│   │   ├── quiz_0_168.nii.gz # mask (0-background; 1-pancreas; 2-lesion)
│   │   ├── quiz_0_168_0000.nii.gz # image
│   │   ├── ...
│   ├── subtype1
│   └── subtype2
├── test # only images are provided
│   ├── quiz_037_0000.nii.gz
│   ├── quiz_045_0000.nii.gz
│   ├── quiz_047_0000.nii.gz
│   ├── ...
```

In `train` and `validation` folders, images and masks are separated based on the subtypes. Each image and mask is named with the format: `quiz_subtype-id_case-id_0000.nii.gz` and `quiz_subtype-id_case-id.nii.gz`, respectively. In the test set folder, only images are provided with the format `quiz_case-id_0000.nii.gz`.

Submission

- `your_name_results.pdf`: A technical report that describes your methods and validation results. Please also release your code on GitHub following the checklist (don't need to share the data) and include the link in your report.
- `your_name_results.zip`: segmentation and classification results of testing cases

```
├── your_name_results.zip
│   ├── quiz_037.nii.gz
│   ├── quiz_045.nii.gz
│   ├── quiz_047.nii.gz
```

```
│   ├── ...
│   subtype_results.csv
```

Please save the classification results as `subtype_results.csv` with the following two columns

Names	Subtype
quiz_037.nii.gz	0
quiz_045.nii.gz	1
quiz_045.nii.gz	2
...	...

Remarks

- Finishing the quiz with the nnUNetv2 framework (`nnU-Net ResEnc M` model) is a mandatory requirement because it provides SOTA segmentation performance and has great flexibilities for network extensions. This code repository is also extensively used in many of our projects.
- For a fair comparison, it is not allowed to use any publicly available datasets or pre-trained weights. Please also don't use the validation set as training set during model development. The role of validation set is to debug or moniter the performance during training.
- The default number of training epoch is 1000 in nnU-Net. Considering the limited runtime on Colab/Kaggle, you don't need to train so many epochs. The model training can be stopped when the loss converges.

Expectations

- Segmentation performance
  - whole pancreas (normal pancreas (label==1) + pancreas lesion (label==2) `np.uint8(label > 0)`) DSC: 0.91+,
  - pancreas lesion DSC (`np.uint8(label==2)`): 0.31+
- Classification performance: macro-average F1: 0.7+
- Reduce the default inference runtime by 10%

Related Work

We highly recommend reading the following papers.

- Isensee, F., Jaeger, P.F., Kohl, S.A.A. et al. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nat Methods 18, 203–211 (2021). https://doi.org/10.1038/s41592-020-01008-z
- Cao, K., Xia, Y., Yao, J. et al. Large-scale pancreatic cancer detection via non-contrast CT and deep learning. Nat Med 29, 3033–3043 (2023). https://doi.org/10.1038/s41591-023-02640-w
- Maier-Hein, L., Reinke, A., Godau, P. et al. Metrics reloaded: recommendations for image analysis validation. Nat Methods 21, 195–212 (2024). https://doi.org/10.1038/s41592-023-02151-z
- Hu, Yujian, et al. "Rapid and Accurate Diagnosis of Acute Aortic Syndrome using Non-contrast CT: A Large-scale, Retrospective, Multi-center and AI-based Study." arXiv preprint arXiv:2406.15222 (2024). https://arxiv.org/abs/2406.15222

- Fast and Low-resource semi-supervised Abdominal oRgan sEgmentation in CT https://flare22.grand-challenge.org/awards/
- Fast, Low-resource, and Accurate oRgan and Pan-cancer sEgmentation in Abdomen CT https://codalab.lisn.upsaclay.fr/competitions/12239#learn_the_details-awards