Technical Report for Deep Learning-based Pancreas Cancer Segmentation and Classification with nnUNetv2

Github link:

https://github.com/AaronC-BME/nnUNetv2_Multi_Task_learning

Methods

Mathematical Setting and Model

The project addresses a multi-task learning problem for pancreas cancer segmentation and lesion subtype classification using 3D CT scans. The model is designed with a shared encoder for feature extraction and two decoders for segmentation and classification tasks.

Network Architecture

Layer Details in Each Block/Module

- **Encoder**:
    - Type: PlainConvUNet
    - Layers: 6 convolutional stages with increasing feature dimensions (32, 64, 128, 256, 512, 512).
    - Non-linearity: LeakyReLU.
    - Normalization: Instance normalization.
- **Segmentation Decoder**:
    - Structure: Mirror architecture of the encoder.
    - Skip connections: Yes.
- **Classification Head**:
    - Layers: Global average pooling followed by a fully connected layer.
    - Final Activation: Softmax. The classification head consists of an adaptive average pooling layer, a flatten operation, and a fully connected linear layer. Softmax activation is applied during the prediction phase.

Number of Parameters

- Total parameters: 25,146,563

- Shared Encoder: 12,572,800
- Decoder: 12,572,800
- Classification Head: 963

Evaluation Metrics

- **Segmentation**:
  - Dice Score.
  - Cross-Entropy Loss.
- **Classification**:
  - Accuracy, Precision, Recall, and F1 Score.

Experiments

Dataset

Dataset was provided as part of the quiz, no other data was used

Dataset Characteristics

The dataset provided contains 3D pancreas CT scans cropped to regions of interest (ROIs) to reduce computational requirements. It consists of:

- **Training Set:**
  - Subtype 0: 62 cases
  - Subtype 1: 106 cases
  - Subtype 2: 84 cases
- **Validation Set:**
  - Subtype 0: 9 cases
  - Subtype 1: 15 cases
  - Subtype 2: 12 cases
- **Test Set:** 72 cases, only images are provided

Each case in the training and validation sets includes a segmentation mask with labels:

- 0: Background
- 1: Pancreas
- 2: Lesion

Preprocessing

Preprocessing was completed by using the nnUNetv2_plan_and_preprocess command

Preprocessing steps include:

- CT Normalization using foreground intensity statistics.
- Resampling to a standard spacing of [2.0, 0.73, 0.73].
- Transposing axes to align with nnUNet conventions.
- Cropping non-zero regions for efficient computation.

Training Protocol

Computing Infrastructure

- GPU: NVIDIA GeForce RTX 3060 Ti
- Number of GPU: 1

Patch Size and Sampling

- Patch size: [64, 128, 192]
- Sampling Strategy: Oversampling foreground regions to balance class distribution

Batch Size: 3

Optimizer and Learning Rate

- Optimizer: SGD with Nesterov momentum.
- Initial learning rate: 0.01.
- Scheduler: Polynomial decay.

Loss Function

- Segmentation: Combined Dice and Cross-Entropy Loss.
- Classification: Cross-Entropy Loss.

Data Augmentation

Augmentation techniques applied during training (part of the nnUNetv2 framework):

- Random rotations and scaling.
- Elastic deformations.
- Gamma transformations.
- Gaussian noise and blur.
- Intensity augmentation for contrast and brightness adjustments.

Training Time

- Approximate time: ~45.8 hours (165 seconds per epoch for 1000 epochs).

Testing Steps

Patch Aggregation: Sliding window approach with Gaussian weighting.

Average inference time per case: ~10 seconds

Results

Validation Results

- Segmentation:
  - Mean Dice Score (Pancreas+Lesion): 0.7526710958310098
  - Dice Score (Pancreas): 0.8937863042679479
  - Dice Score (Lesion): 0.6115558873940716
- Classification:
  - Accuracy: 0.75
  - Precision: 0.7652116402116403
  - Recall: 0.75
  - F1_score: 0.7430555555555556
  - Macro-averaged_F1_score: 0.7388888888888889
  - Confusion Matrix: [6, 2, 1],

    [0, 14, 1],

    [1, 4, 7]

Strategies to improve performance

The following are strategies to improve the performance since segmentation performance of pancreas (Label 1&2) did not reach 0.91 dice score, however, the model did meet the criteria for both lesion segmentation and classification performance, reaching 0.61 dice score and 0.738 macro average F1 score respectively.

1. Target Fine-Tuning
   a. Freezing part of the network (classification head) and train specifically for segmentation
2. Learning rate adjustment
   a. Adjusting learning rate schedule to decrease more performance seems to have plateau
3. Class balancing
   a. Adjusting class weights in the loss function or oversample underrepresented class in the dataloader
4. Adding to the model architecture:
   a. Adding to the model architecture generated from nnUNetv2_plan_and_preprocess command. Such as adding regularization layers to improve generalization or more expressive decoder to improve segmentation accuracy.