UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Topic: Image Processing of Multiple datasets for 3D Medical Image Analysis

Submitted By

Aayush Manoj Tirmalle

4730148

Guided By

Wenzhao Zhao

Contents

Abstract

This report presents the results of preprocessing multiple 3D medical datasets for medical image analysis tasks, including medical image segmentation and classification. Two public software tools from GitHub repositories are studied: nnUNet, designed for biomedical image segmentation, and a deep learning model for early Alzheimer's disease detection using structural MRIs. The objective is to verify and test the effectiveness of the preprocessing pipelines from these software tools. Specifically, we analyze the design and training process of nnUNet and evaluate its performance on various biomedical datasets. For the Alzheimer's disease detection task, we train and evaluate a deep learning model.

Introduction

Medical image analysis plays a crucial role in healthcare, aiding in diagnosis, treatment planning, and disease monitoring. With the growing availability of medical imaging data, deep learning techniques have become increasingly important in automating the analysis process, particularly in tasks like image segmentation and disease classification. In this report, we explore the preprocessing and training of two advanced deep learning tools for medical image analysis. The first tool, nnUNet, is a self-configuring deep learning-based framework that has become a benchmark for biomedical image segmentation tasks. The second tool is a convolutional neural network (CNN) designed for early detection of Alzheimer's disease using structural MRI data. The primary objective of this report is to verify and evaluate the effectiveness of the preprocessing pipelines used by these tools. nnUNet is tested on a variety of biomedical datasets, and the Alzheimer's CNN model is trained and evaluated to assess its accuracy in detecting early-stage Alzheimer's disease.

Software Overview

Software 1 - nnUNetv2

nnUNet is a deep learning-based segmentation method that automatically configures itself, including preprocessing, network architecture, training, and post-processing for any new task. The key design choices in this process are modeled as a set of fixed parameters, interdependent rules, and empirical decisions. Without manual intervention, nnUNet surpasses most existing approaches, including highly specialized solutions, on 23 public datasets used in international biomedical segmentation competitions.

Software 2 - Alzheimer's Disease Detection

This project involves the design and implementation of a convolutional neural network (CNN) for Alzheimer's disease detection. The model leverages instance normalization, avoids early spatial downsampling, widens layers for improved accuracy, and incorporates age information to enhance predictive performance. Compared to traditional models, the CNN demonstrates higher efficiency and speed, providing a robust solution for early detection and progression forecasting of Alzheimer's disease.

Datasets used

Dataset used for training nnUNet

The datasets used for nnUNetv2 training in this study include:

- **BraTs2021 (Brain Tumor Segmentation)**: This dataset consists of multi-institutional, multi-parametric MRI scans. It includes imaging modalities such as T1, T2, and FLAIR, with annotated tumor sub-regions.
- **AMOS22 (Abdominal Multi-organ Segmentation)**: The AMOS (Abdominal Multi-organ Segmentation) dataset includes abdominal CT scans aimed at segmenting multiple abdominal organs. Organs such as the liver, spleen, pancreas, and kidneys are annotated.

- **KiTS23 (Kidney Tumor Segmentation)**: This dataset contains CT scans of kidney cancer patients, with annotations for the kidney and kidney tumor regions.
- **BTCV (Beyond The Cranial Vault Segmentation)**: The BTCV dataset consists of CT scans aimed at the segmentation of various abdominal organs. Organs include the spleen, pancreas, gallbladder, kidneys, and liver.

Dataset used for Alzheimer's Disease Detection

The ADNI dataset is a comprehensive collection of longitudinal neuroimaging data aimed at understanding Alzheimer's disease progression. It includes various types of imaging data, genetic information, and clinical assessments from subjects diagnosed with Alzheimer's disease, mild cognitive impairment (MCI), and healthy controls. The primary imaging modality in this context is T1-weighted structural MRI, which provides high-resolution images of brain anatomy.

Data Preprocessing pipelines

Data preprocessing pipeline for nnUNetv2

The nnUNetv2 framework performs several preprocessing tasks to prepare the data for effective training in medical image segmentation. Below are the necessary steps:

- Dataset Analysis and Fingerprinting - nnUNetv2 analyzes the dataset to extract characteristics such as voxel spacing, intensity ranges, and label distributions. This information, called the "dataset fingerprint," is crucial for determining the best preprocessing and model configurations, as it identifies dataset-specific properties like resolution and scale.
- Resampling - Voxel Spacing Standardization: nnUNetv2 uses resampling techniques to adjust voxel spacing, ensuring that images across the dataset share consistent spatial resolution. This process is particularly important when datasets come from different sources with varied resolution settings, as it harmonizes voxel dimensions across the training data, which is essential for uniform model performance.
- Normalization - nnUNetv2 applies z-score normalization to standardize image intensity values across all cases. This involves subtracting the mean and dividing by the standard deviation of each image. This step ensures consistency in intensity, making it easier for the model to learn features without being affected by varying brightness or contrast levels among different images.
- Data Augmentation - To enhance the diversity of the training data and improve the model's generalization ability, nnUNetv2 applies real-time data augmentation. Typical augmentations include random rotations, scaling, elastic deformations, and intensity variations. These modifications introduce variability, making the model more robust to new data.
- Dataset Splitting and Cross-Validation - nnUNetv2 splits the dataset into training, validation, and test sets, typically in an 80-20 or 70-30 ratio. It further divides the training set into multiple folds for cross-validation (usually 5). This approach ensures that each data point serves as a validation case at least once, leading to a more balanced and reliable evaluation.
- Configuration Generation - Based on the dataset's fingerprint, nnUNetv2 generates configurations for the most appropriate U-Net architectures:
  - 2D U-Net: Primarily used for datasets with 2D images but can also work for certain types of 3D images.
  - 3D Full Resolution U-Net: Processes 3D datasets at the full available resolution, ideal for detailed segmentation.
  - 3D Low Resolution to 3D Cascade Full Resolution U-Net: A two-step approach for large, high-resolution datasets where an initial low-resolution segmentation is refined by a subsequent high-resolution model, reducing computation load while maintaining accuracy

Data preprocessing pipeline for Alzheimer's Disease

The data preprocessing pipeline for Alzheimer's disease detection using MRI involves several key stages. Below are the main steps implemented in this:

- Data Standardization to BIDS Format: The pipeline begins by converting MRI data to the Brain Imaging Data Structure (BIDS) format using the Clinica framework. This standardized format organizes files, metadata, and anatomical information systematically, making it easier to apply consistent preprocessing steps and manage large datasets effectively.
- Image Preprocessing with Clinica -
  - Skull-Stripping and Spatial Alignment: The MRI images undergo skull-stripping to remove non-brain tissues, enhancing the focus on brain structures. Spatial alignment registers images into a common coordinate space, ensuring consistent orientation and reducing variability between scans.
  - Normalization and Intensity Adjustment: Clinica normalizes intensity values across MRI images, often through z-score normalization, to correct for brightness variations. This helps the model generalize well across different scans and imaging setups.
- Data Splitting and Organization: The pipeline organizes the data into training, validation, and test sets to allow unbiased evaluation and model tuning. Each subset undergoes consistent preprocessing to maintain a standard input format across all stages.
- Optional Data Augmentation: The framework optionally supports data augmentation during training. Techniques such as rotations, flips, scaling, and intensity adjustments introduce variability to the dataset, improving the model's robustness by simulating different imaging conditions.
- Configuration and Parameter Generation: The preprocessing pipeline generates configuration files that define parameters for model architecture, data handling, and optional use of patient age as a feature. This structured configuration enables easy adjustments for model-specific needs.

Data Processing Steps

Data processing steps for nnUNetv2

- Data Preparation -
  - Dataset Sourcing: Public datasets are downloaded from repositories and structured in "nnUNet_raw" in a specific format. Each dataset should be named in the format DatasetXXX_NAME (e.g., Dataset001_Liver).
  - Directory Structure: Within each dataset folder, create subdirectories: "imagesTr" for training images (usually in NIfTI format, .nii.gz). "imagesTs" for test images. "labelsTr" for ground truth segmentation labels for training data.
  - Dataset Metadata: The "dataset.json" file within each dataset directory specifies essential metadata like label mappings, modality information, and other configuration details needed for nnUNet to interpret the dataset.
- Preprocessing - Run the preprocessing command, "nnUNetv2_plan_and_preprocess", to standardize the dataset. This includes steps like:
  - Intensity Normalization - Adjusts intensity values to make datasets comparable across scans.
  - Resampling - Resizes images to match the resolution required by the selected U-Net architecture (2D, 3D full resolution, etc.), ensuring uniformity in spatial dimensions.
- Output Storage - Preprocessed data is stored in the "nnUNet_preprocessed" directory. This output is structured for efficient access during training.
- Cross-Validation Splits -

  - k-Fold Cross-Validation - The dataset is split into multiple folds (typically 5) for cross-validation. In each round, a different subset (fold) is used as the validation set while the remaining folds serve as training data.
  - Fold Usage - This ensures that each data point is used for validation once, leading to a more robust model by exposing it to all data through both training and validation.
- Dataset Loader Configuration -
  - Data Loading Scripts - nnUNetv2 includes Python scripts to prepare data loaders that dynamically load and preprocess data batches during training. This includes handling augmentations, such as rotations and scaling, to increase data diversity.
  - Automatic Model Configuration - nnUNetv2 analyzes each dataset's characteristics and automatically configures appropriate network settings and parameters based on the "dataset fingerprint" (dataset-specific attributes like spacing, modality, and resolution).
- Model Configuration - Based on the dataset properties, nnUNetv2 will select or suggest specific model configurations, like 2D or 3D U-Nets, and adjust relevant hyperparameters. This automated selection improves model performance by tailoring the architecture to the dataset.

Data processing for Alzheimer's Disease Detection

- Data Conversion to BIDS Format - The conversion of MRI scans into BIDS format is performed using Clinica commands, which can be automated through a shell script by running "run_convert.sh". This command "clinica convert adni-to-bids" ensures that all files are organized uniformly, with metadata and image files stored in a structured directory layout.
- Data Preprocessing - The repository contains a set of preprocessing scripts, each handling specific stages of data preparation:
  - run_convert.sh: This script executes the initial data conversion to BIDS format.
  - run_adni_preprocess.sh: Processes the primary ADNI dataset for training, focusing on normalizing, skull-stripping, and spatially aligning the MRI images.
  - run_adni_preprocess_val.sh and run_adni_preprocess_test.sh: These scripts separately preprocess the validation and test data subsets. This division maintains consistency in data preparation across all stages, which is critical for reliable model evaluation.
- Data Splitting - The dataset is split into training, validation, and test sets. This is done in a way that each subject appears in only one of the sets, preventing data leakage and ensuring fair model evaluation.
- Final Preparation - The final preprocessed files are organized in the outputs folder in a way that's compatible with the model training scripts in the repository. This preparation step ensures seamless integration between data loading and model training.

Training procedure

Training procedure for nnUNetv2

The nnUNetv2 framework automates the training pipeline based on dataset characteristics, optimizing hyperparameters and model architecture.

Training workflow:

- Selecting Model Configuration: Based on the dataset fingerprint established during preprocessing, nnUNetv2 automatically chooses the most appropriate model configuration. Options include 2D, 3D full resolution, or a cascaded approach using both low and full resolutions for larger 3D datasets.

This configuration ensures that the model architecture matches the dataset's spatial characteristics, improving segmentation accuracy.

- Running the Training Command:
  - Training Command: Start the training process using the command: "nnUNetv2_train <Dataset_name_or_ID> <unet_configuration> FOLD –npz"
    - Parameters:
      - Dataset_name_or_ID: Identifies the dataset to be used for training.
      - unet_configuration: Specifies the model configuration (e.g., 2d, 3d_fullres).
      - FOLD: Indicates which fold to train on for cross-validation, typically ranging from 0 to 4 for 5-fold cross-validation.
      - Saving Predictions: Using the –npz flag stores softmax predictions for each validation fold, useful for ensemble methods or later analysis.
- Logs and Outputs: Training logs, metrics, and model weights are saved in "nnUNet_results" within subdirectories organized by dataset, model configuration, and fold. These logs document training progress, including loss reduction, Dice coefficient scores, and other relevant metrics.
- Training with Multiple Folds: nnUNetv2 performs training across multiple folds (e.g., 5-fold cross-validation). This allows each subset of the data to be used for validation, providing a balanced and comprehensive assessment of the model's generalization ability. After completing training on all folds, the results can be averaged or ensembled for better performance.
- Automating Cross-Validation: Running nnUNetv2_train iteratively on each fold ensures that every data point is utilized for both training and validation, producing a robust model performance estimate.
- Hyperparameter Tuning: nnUNetv2 self-adjusts key hyperparameters, such as learning rate, batch size, and patch size, based on the dataset fingerprint. These adjustments optimize the model's performance without requiring extensive manual tuning

Training procedure for Alzheimer's Disease Detection

- Configuring Model Parameters: Model parameters, such as the widening factor, age inclusion, and batch size, are defined in configuration files. This flexibility allows researchers to experiment with different model architectures and configurations tailored to their specific needs.
- Initiating Training: Training is conducted via the main.py script, which initiates the training process with the specified configuration. The "config_file.yaml" includes all relevant settings, such as learning rate, batch size, and data augmentation strategies.
- The following hyperparameters were used during training:
  - Batch Size: 4 for training and 2 for validation.
  - Number of Epochs: 50.
  - Optimizer: Stochastic Gradient Descent (SGD).
  - Learning Rate: 0.01, with no weight decay.
  - Model Architecture:
    - Input Channel: 1
    - Hidden Layer Size (nhid): 512
    - Feature Dimension (feature_dim): 1024
    - Expansion Factor: 8

    - Number of Labels: 3 (CN, MCI, AD)
    - Normalization Type: Instance Normalization
    - Convolutional Layer Type: conv3x3x3
  - Training Iterations: Maximum of 16,000 iterations.
- Evaluating the Model: The "Model_eval.ipynb", a Jupyter Notebook specifically designed for evaluating the model's performance on the test set. This notebook loads the trained model checkpoints, runs predictions on the test data, and computes performance metrics, including accuracy, sensitivity, and specificity. The notebook provides visualization tools to analyze misclassifications and understand model limitations.

Results of Evaluation of Models

nnUNetv2

The following graph presents the training and validation loss, along with the pseudo-Dice coefficient, for nnUNetv2 trained on the AMOS22 dataset, using a PlainConvUNet architecture with 8 stages. The model was trained with a batch size of 1 and an input image size of 640 × 640 pixels. The optimizer used for training was Adam with an initial learning rate of 1e-2, which was later reduced to 2e-05 in later epochs. The loss function employed was a combination of Cross-Entropy (CE) Loss and Dice Loss, ensuring better segmentation performance by handling class imbalances effectively.

During training, the training loss (loss_tr) and validation loss (loss_val) showed negative values, which may be due to a custom implementation of the Dice loss function. The final recorded values were train_loss = -0.156 and val_loss = -0.151 at epoch 999. These decreasing loss values indicate effective model convergence. The pseudo-Dice coefficient, which reflects segmentation accuracy across different anatomical structures, increased over time to 0.93, training time per epoch averaged around 11.7 seconds.

Although nnUNetv2 was trained on multiple datasets, including BraTS2021, KiTS23, BTCV, BraTS23, and KiTS19, only the results for the AMOS22 dataset are shown here. The plot demonstrates the model's learning progression over 1000 epochs, with the training loss (loss_tr) and validation loss (loss_val) decreasing over time, indicating effective model convergence. The pseudo-Dice coefficient, which reflects segmentation accuracy, also increases, showcasing the model's improvement during training.

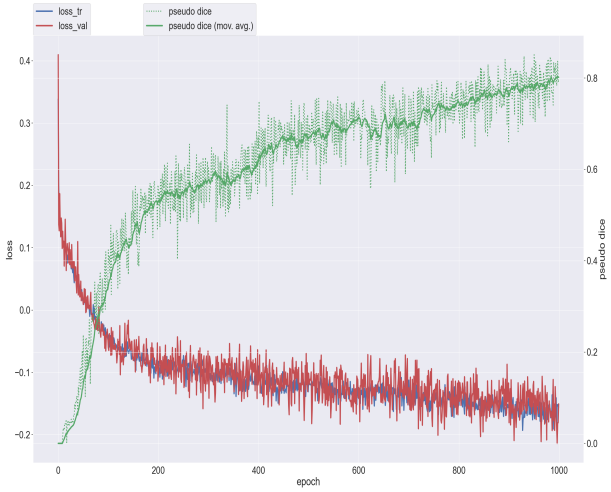

Figure 1: Model Performance Metrics: Loss and Pseudo-Dice Scores Across Training Epochs

Alzheimer's Disease Detection

The CNN model trained for Alzheimer's disease detection using MRI data was evaluated on three classification tasks: differentiating between cognitively normal (CN), mild cognitive impairment (MCI), and Alzheimer's disease (AD) cases. The model's performance was assessed using ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) metrics, which measure the model's ability to distinguish between the conditions. Below is a summary of the results:

CN vs. All: The ROC-AUC score for distinguishing CN (cognitively normal) individuals from other categories (MCI and AD) is 0.86. This score suggests that the model performs well in identifying cognitively normal individuals, showing a high true positive rate relative to the false positive rate for this category.

MCI vs. All: For MCI vs. all, the ROC-AUC score is 0.47. This relatively low score indicates that the model struggles to accurately differentiate MCI cases from CN and AD. This could be due to the subtle structural brain changes associated with MCI, which are harder to capture in MRI data compared to changes seen in AD.

AD vs. All: The ROC-AUC score for distinguishing AD from other categories is 0.86. This suggests that the model is effective at identifying AD patients, likely due to more pronounced brain structural differences in Alzheimer's cases that the CNN can learn and recognize.

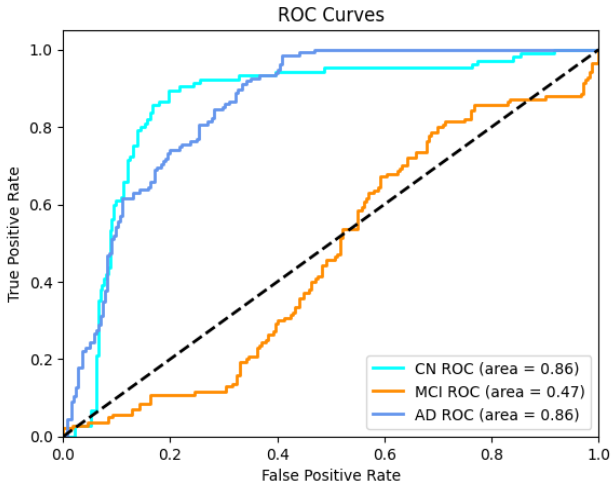

Figure 2: ROC Curves for Alzheimer's Disease Detection Model.

Conclusion

This study focused on testing and evaluating the data preprocessing pipelines implemented in nnUNetv2 for medical image segmentation and a deep learning model for Alzheimer's disease detection. The objective was to assess the effectiveness of these preprocessing pipelines and their impact on model training and evaluation.

For nnUNetv2, multiple datasets were preprocessed using the framework's automated pipeline, which includes resampling, intensity normalization, and data augmentation. The performance of the segmentation model was evaluated using the pseudo-Dice coefficient, demonstrating effective convergence over training epochs.

For Alzheimer's disease detection, MRI data underwent preprocessing using the Clinica framework, including skull-stripping, intensity normalization, and spatial alignment. The preprocessed data was then used for model training, and classification performance was assessed using ROC-AUC scores.

The results confirm that both preprocessing pipelines effectively standardize and optimize medical imaging datasets for deep learning applications. This evaluation serves as a validation of the preprocessing approaches implemented in these frameworks.

Github repository

https://github.com/Aayushtirmalle/Image_Processing_of_Multiple_datasets_for_3D_Medical_Image_Analysis

Reference

1. https://github.com/MIC-DKFZ/nnUNet/tree/master
2. https://github.com/NYUMedML/CNN_design_for_AD