FINAL PROJECT PROPOSAL - SAT 5114 – AI IN HEALTHCARE

Project Members - Nagendra Madi Reddy & Vishwajeeth Balaji

Project Title - Visualizing Health - Image Segmentation for Brain Tumor Detection

Project Contributions

Vishwajeeth - Worked on the data and performed all the data preprocessing and dataset management along with the image augmentation to maintain the robustness and accuracy of the training data. Overall worked on data cleaning to till training and evaluation.

Nagendra - Worked on the model evaluation, optimization and model development using deep learning techniques. Eventually projected the final design and implementation of U-Net architecture and tailored it for final medical image segmentation.

Image Classification & Semantic Segmentation

Brain MRI scans are categorized into many groups and locations in order to identify tumors in the images. To do this semantic segmentation is carried out to separates MRI scans into discrete sections according to their semantic content and is then used to accomplish this categorization, which helps with the precise identification of abnormalities like tumors.

Clinical Problems

Tumor Identification

The project primary goal is to accurately identify the brain tumors within the MRI scans. Identify the tumors at an early stage helps clinicians to provide proper treatment and improve patient outcome.

Tumor Mapping

Mapping the pin-point locations of the tumor and length of the tumor in the brain will help in carrying out proper surgical procedures and target the location when performing radiotherapy.

Sub-type Specific Tumor Segmentation

Different tumors show different characteristics and behaviours, so it’s important to classify the sub-types of the tumors to provide personalized treatment/medicine to the patient.

Significance

Enhanced Diagnostic Accuracy

Proper and precise classification/segmentation of the tumors in the brain can help clinicians and other radiologist to make informed decisions and further provide accurate treatment & patient management.

Enhanced Treatment Design

Accurate segmentation of the tumors within the brain can help to visualize the tumor limits and its interaction with the surrounding brain areas. This in-turn helps surgeons and radiologists to carry out surgical resection and radiation therapy.

Enhanced R&D Initiatives

Developing automated segmentation methods can help to drive in analysing large data volumes of the MRI scans / medical images, thereby bringing research into better biomarkers, treatment and understanding disease patterns.

Decreased Subjective Bias

Automated segmentation provides more accurate & consistent tumor detection when compared to manual segmentation. Moreover manual segmentation is time-consuming process.

Enhanced Patient Prognosis

Enhancing precise tumor segmentation & detection can improve patient health by increasing survival rates, reducing treatment complications, and finally improving overall patient condition.

CONVOLUTIONAL NEURAL NETWORK MODEL

The motive of the semantic segmentation is to properly detect the tumors in the brain for evaluating clinical problems, tumor localization and finally suggest proper treatment plan along with research in neuro – oncology.

Data Acquisition and Cleaning

Gather MRI scans with brain images having tumors and without tumors.

Use the brain scan images to preprocess which includes pixel normalization, image resizing and data argumentation to increase disparity of the training set.

Data Distribution

The data is distributed into training , validation, and testing. The training set is used to train the model on the data, validation set is used to perform the hyperparameter tuning and check the performance during training and finally the testing set is used to monitor the overall model performance on unseen data.

CNN Model Design

Develop a U-Net model for semantic segmentation of medical images/ scans. It is widely accepted for its effectiveness and analysis.

The following are the U-Net model components:

- **Contracting Path (Encoder):**

Through convolutional and pooling layers, the contracting path, also known as the encoder, increases feature channels while decreasing the size of the input image. ReLU activation is applied after each convolutional layer to achieve non-linearity. In order to obtain high-level features, max-pooling layers downsample, and at each stage, double the feature map depth to capture intricate patterns.

- **Expansive Path (Decoder):**

The expanding route, also known as the decoder, uses convolutional layers and upsampling to restore spatial resolution. Spatial dimensions are increased using upsampling techniques like bilinear interpolation and transpose convolution. In order to maintain spatial information and enhance gradient flow, skip connections link corresponding feature maps from the contracted path. Prior to additional convolutional computations, feature maps from both routes are combined using concatenation.

- **Bottleneck:**

The expanding and shrinking pathways are connected by the bottleneck layer. To capture rich semantic information, it usually consists of several convolutional layers with a large number of feature channels. While serving as a bottleneck for geographical data, the bottleneck layer enables the network to capture intricate details.

- **Final Layer:**

A convolutional layer and an appropriate activation function—such as the sigmoid function for binary classification tasks or the softmax for multi-class classification tasks—usually make up the last layer of the U-Net model. Each pixel in the input image's expected segmentation mask is represented by the final layer's output.

Model Training

Compile the model with the necessary loss functions and optimizer such as binary cross-entropy and adam optimizer.

Train the model using training set and validate the performance with respect to validation set.

Based upon the callbacks detect and stop overfitting to save the best model with respect to validation performance.

Model Evaluation

Analyze the trained model using the training set and evaluate the performance metrics such as accuracy, precision, f1score & recall.

To assess the performance qualitatively, make visuals that contrast the model's predictions with the real ground truth masks.

Model Deployment

After reaching satisfactory results from the model deploy it on new MRI images

Segment the brain tumor images using trained model on the unseen MRI scans, therefore providing insights to the doctors/clinicians for diagnosis suggestion and treatment planning.

Proper execution of the above U-Net model for semantic segmentation we can achieve effective CNN model for detecting the brain tumors in MRI scans.

CNN MODEL HYPERPARAMETERS

The CNN Model that is developed has many hyperparameters that plays a vital role in determining model performance.

The following are the hyperparameters used for tuning the sets:

Hyperparameters for U-Net Architecture:

Pooling Operations: The type of pooling operation used is the max pooling. Often max pooling with 2x2 windows.

Activation Functions: Mostly used is ReLU, but alternatives like ELU or Leaky ReLU can be effective for each convolutional layers.

Dropout Rate: It helps in preventing overfitting, commonly ranging values from 0.1 to 0.5. It is defined as way to find hidden networks during training

Batch Size: Determined by hardware constraints and dataset size. It is a process of sample processing based upon the iterations.

Loss Function: This function is used to measure the difference in predicted and ground truth values. Chosen based on task characteristics (e.g., binary cross-entropy for binary segmentation).

Optimizer: Adam, SGD, or RMSprop are the some of the options for optimization. Mostly Adam is preferred when compared to other optimizers.

Learning Rate: Often set to a small value (e.g., 0.001) and adjusted during training. It is defined as the rate at which model parameters are updated during optimization.

Convolutional Layers: Determine based on task complexity and dataset size. These are the layers chosen in encoder and decoder of the U-Net model.

Filter Sizes: Typically 3x3 or 5x5, these are the default filter sizes used in the convolutional layers.

Number of Filters: Commonly increasing in powers of 2 (e.g., 16, 32, 64, 128). These are the filters present in each convolutional layers which determines the feature map depth.

Hyperparameter Tuning Techniques:

Grid Search or Random Search: Investigate the preset hyperparameter space.

Cross-Validation: Evaluate your ability to generalize across various subsets of training data.

Manual Tuning: Make hyperparameter adjustments and experiments based on your intuition and domain expertise.

By incorporating the hyperparameter tuning and validation techniques we can fine tune the CNN model to achieve the best results in detecting the brain tumor.

METHODS TO PREVENT OVERFITTING

The below mentioned techniques are used to prevent overfitting in the CNN model

Early Stopping: By keeping an eye on performance on a different validation set, stop training when validation loss becomes stagnant in order to avoid overfitting.

Weight Regularization: To discourage overfitting and encourage simpler patterns, penalize big weights using L1/L2 regularization.

Data Dropout: To replicate noisy data and increase model robustness, randomly zero out input characteristics during training.

Data Augmentation: To improve generalization, add transformations such as rotations, translations, flips, and zooms to increase the diversity of training data.

Dropout: During training, randomly deactivate neurons to regularize the network by promoting the acquisition of robust characteristics and minimizing dependence on particular neurons.

Batch Normalization: To improve stability and generalization, normalize layer activations to reduce internal covariate shift problems.

Ensemble Methods: Train several CNN models with different architectures or initializations, then combine the predictions to improve generalization and reduce overfitting.

Data Preprocessing: To stabilize training and avoid overfitting, optimize the normalization, scaling, and feature engineering of input data.

Reduce Model Complexity: To counteract overfitting, reduce the number of layers, neurons, or other components in the architecture.

Data stratification: To avoid learning erroneous correlations, make sure that the class distributions are constant between the training, validation, and testing sets.

The dataset is taken from the Kaggle repository and below is the link to the dataset.

https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection