Software Requirements Specification

for

Prediction of survival rate for patients with brain tumor using radionomic features and machine learning.

Guide: Prof. Disha S . Wankhede

Sourav Singh
Shashank Gopinath Menon
Faraz Usman Karjikar

Vishwakarma Institute of Information Technology, Pune

Version 1.0

1. Introduction

1.1 Purpose

This document provides a detailed description of Software Requirements specification(SRS) for Prediction system. It is prepared according to "IEEE Recommended Statements for Software Requirements Specifications- IEEE Standard 830 - 1998".
The Software requirements specification (SRS) document is intended to provide the requirements of prediction system project and the expectations of the stakeholders of the project.The document includes the project perspective, data model and constraints of the overall system.

The intended audiences of the document are project managers, developers and end users.

- Project managers will review the document and determine whether the planned system fulfills the requirements.
- Developers provide consistency by using the documentation.
- End users use the document to learn about the scope of the project and its capabilities.

1.2 Intended Audience and Reading Suggestions

This document is mainly intended for the developers, the project guide and the experts. This document consists the software requirements specification for our project i.e. Prediction of survival rate for patients with brain tumor using radionomic features and machine learning, starting from introduction, then features and ending with functional and non-functional requirements. The document will require knowledge of anatomy and biology to an extent, which will be explained if and when needed.

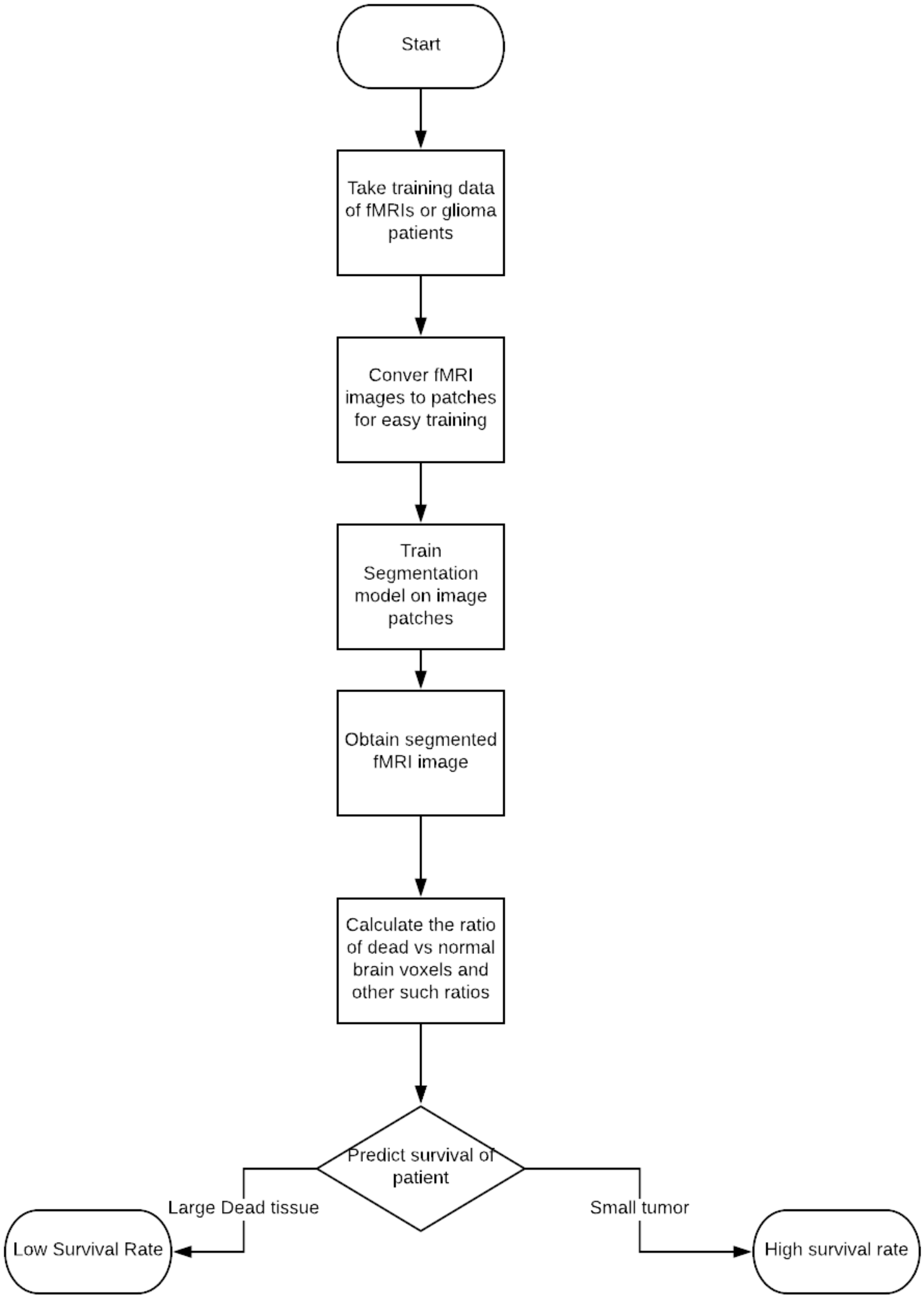
1.3 Existing Applications

The prediction of overall survival of a patient afflicted with glioma (general term for brain tumor) is one of the important research problems and is a part of the MICCAI BraTS Challenge, where researchers aim to create a model which can quantify the overall survival of a patient with brain tumor in the best way possible. The participants belong to various institutions like NUS, Tencent, SUSTech, DKFZ and many others. Current entries to the competition can be seen herehttps://www.cbica.upenn.edu/BraTS18/ and implementations from last year can be read here- https://www.cbica.upenn.edu/sbia/Spyridon.Bakas/MICCAI_BraTS/MICCAI_BraTS_2017_proceedings_shortPapers.pdf.

2. Functional Specification

2.1 Project Perspective

Traditionally, quantifying overall survival for a patient with brain tumor would be difficult for a doctor or a radiologist, since it required experience in reading the fMRI scan of the patient and understanding it nicely. While doctors and radiologists can read the fMRI well, they sometimes can read too much into an fMRI and can sometimes misdiagnose the survival rate of a patient with a small tumor. The system should allow for reading the fMRI data of the patients with brain tumor and provide the overall survival rate of these patients based on how much percent of the tissue has turned necrotic(dead), tumor regions with edema etc. The diagram for the pipeline is as follows-


Start
Take training data of fMRIs or glioma patients
Conver fMRI images to patches for easy training
Train Segmentation model on image patches
Obtain segmented fMRI image
Calculate the ratio of dead vs normal brain voxels and other such ratios
Predict survival of patient
Large Dead tissue
Small tumor
Low Survival Rate
High survival rate

3. Scenarios

The user will give the training data required to train the Neural Network as an input and then the fMRI images will be preprocessed and neural network will be run to gain information on the overall survival of the patient. Basic steps behind the pipeline are-

1) Input the training and validate the format of the training data.
2) Convert the images to image patches for easier training of neural network.
3) Train the neural network on the image patches.
4) Obtain segmentation of brain tumor from image patches.
5) Calculate necrotic vs normal voxel ratios, tumor vs normal voxel ratios. These will be required to calculate the survival of the patient.
6) Use the features calculate at 5) to get the overall survival of a patient.

3.1 User classes and characteristics

The project will be divided into two classes- segmentation and survival-prediction. The segmentation class will deal with preprocessing of training data, obtaining image patches and getting segmentation of brain tumor. Survival-prediction class will deal with obtaining ratios for normal vs necrotic voxels and tumor vs normal voxels and use these as features to calculate the survival rate of a patient.

3.2 Design and Implementation constraints

The main constraint lies in getting a better understanding of the data and how the distribution of the tumor lies in the fMRI images. Another constraint lies in selecting the best algorithms for segmenting the brain tumor from images, calculating features and then predicting the survival rate of the patients.

3.3 Assumptions and Dependencies

The prediction systems assumes that there is a training data present from where it needs to learn from. Without the training data, the further steps in the pipeline will not work.

4. External Interface Requirements

4.1 User Interfaces

Since the project delves into the research aspect, the user interface will be a commandline interface from where the user can set the input path for training data, batch size, learning rate and various other parameters for the learning of neural networks. The project will make use of Object-Oriented programming and abstract the code into various modules and classes for easy usability and testing.

5. Technical Specification

5.1 Operating Environment

The Operating system will be Debian Stretch with CUDA 9.0 and cuDNN 5.0 for acceleration of neural network training. The coding will be done using Python, with occasional visualizations done in R and stored as R Markdown. The neural network architecture will be written in Tensorflow and PyTorch and the survival predictor will make use of LightGBM for prediction purposes. The hardware will be Intel Skylake Xeon E3 with 8 virtual cores and 8 physical cores, 30 GB RAM, 500 GB Hard Disk and Nvidia Tesla K80 with 24 GB of GDDR5 memory.

5.2 Programming Language and technologies

1) <u>Python-</u> The programming language will be used to write the main programming logic for the prediction system.
2) <u>R-</u> The language will be used for visualizations of fMRI images and to analyse the data.
3) <u>Tensorflow-</u> The Google product will be used for quickly prototyping the neural network archictecture and train the neural network on the training data to obtain segmentations of the fMRI images.
4) <u>PyTorch-</u> PyTorch is a second option for prototyping of neural networks, with its use of automatic differentiation to allow for a define-by-run framework.
5) <u>LightGBM-</u> The Microsoft product will be useful for predicting the survival rate of the glioma patients because of its high-performance and support for GPU learning.
6) <u>CUDA and cuDNN-</u> The libraries will be important since they can help accelerate training of neural networks by upto 3 times, which would otherwise take a lot of time.

5.3 Performance Constraints

The performance constraints of the system would depend on the depth of the neural network and the size of the training data. If the depth of the neural network is large, the neural network would take time to iterate through a set of images and learn features from it. So, the time of the algorithm would depend on the depth of neural network and the complexity of the predictor. If the predictor being used is based of Ensemble Stacking, it will take time for the predictor to learn and then give and output.