Software Requirements Specification

For

Detection of Diabetic Retinopathy and Thyroiditis using Artificial Neural Networks

Version 1.0

Prepared by :

Chaitrali Londhe
Anirudh Murli
Pranav Acharekar
Pranav Bahulkar

VIIT

16/09/2018

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to build an automated autoimmune disease detection system for Diabetic Retinopathy and Thyroiditis.

1.2 Intended Audience and Reading Suggestions

This document is intended for project guide and experts. This project is useful for doctors. Section 1.4 briefs about the scope of project, section 2 walks you through overall description of product like functions, operational environment,design and implementation constraints,etc. Interface requirements is described in section 3. System features are listed in section 4 and section 5 lists non functional requirements.

1.3 Product Scope

Our goal is to generate automatic method for detection of diabetic retinopathy and thyroiditis which will assist clinician. The required accuracy and timing of this care which is of significant importance to both the cost and effective-

ness of treatment could be achieved. Early detection of autoimmune diseases is a time-consuming process even for a well–trained clinician, which may result in delayed treatment, miscommunication, etc.

1.4 References

https://ieeexplore.ieee.org/document/4809154/

https://ieeexplore.ieee.org/document/1199031/

https://www.sciencedirect.com/science/article/pii/S1877050916311929

https://dzone.com/articles/8-best-deep-learning-frameworks

https://machinelearningmastery.com/check-point-deep-learning-models-keras/

https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/

2. Overall Description

2.1 Product Perspective

The product is designed for improving the diagnosis process of Hashimoto's thyroiditis and Diabetic Retinopathy. It would serve as a tool for doctors to speed up the diagnosis process without any compromise on the precision.
This would be a completely new, self-contained product. This documents explains the complete system and its required components. A simple component diagram is shown below. The description of each component is given in section 3.

2.2 Functional Specification

2.2.1 System Overview

The system is primarily designed to improve the accuracy and time required to diagnose Hashimoto’s thyroiditis and Diabetic Retinopathy. It is designed in simplistic manner and can be used very easily without any special kind of training.

General Steps to use the system:

1. The user can choose the disease for which he wants to carry out detection.
2. In case of thyroiditis, the user is shown a page where he needs to enter the required values of patient in text format. For diabetic retinopathy, the user will have to upload scans of the retina in the system.
3. After the above step, user will click a button to get the required results.
4. The results are then verified by the experts.
5. The system will periodically train itself on the verified results to improve its accuracy.

The user machine requires python interpreter with required libraries installed in it.

2.2.2 Scenarios

The primary users of the system would be doctors and clinicians.

Scenario for Thyroiditis :

Suppose Dr. Rahul want to know whether Amit, his patient, has Hashimoto’s thyroiditis or not. Rahul looks at the test reports and enter the values in the system. Then simply he clicks on a button to get the result.

1. The system shows whether Amit is having Thyroiditis or not. If found positive, the system also tells whether it is hypothyroidism or hyperthyroidism. This way Dr. Rahul is able to obtain results in very less time. To verify the results, Rahul will only need to look for certain symptoms which confirms the presence of the disease.
2. In case if Amit does not suffer from Hashimoto's thyroiditis, the system will simply show result as negative.

Dr. Rahul can will gain more confidence in the system after using it for some cases as the system is trained using his own diagnosis results.

Scenario for Diabetic Retinopathy :

Let us consider a situation in which Dr. Rao, a Diabetic Retinopathy Expert, interacts with almost 60-70 patients a day. He gets tired after continuously checking patients. So after about 40-45 patients the only thing on his mind is going home and falling asleep.

However, one evening, a patient Mr. Jay has come in with blurry vision. He is the last patient of the day. Dr Rao can only think about falling asleep. But negligence on his part might cause him trouble. With some efforts, he rubs the sleep out of his eyes and examines Mr. Jay.

He suspects that Mr. Jay might be suffering from Diabetic Retinopathy given that he is a patient of type 1 Diabetes. He scans Jay's retina and uploads these images into the system and clicks a button to get results.

1. The system will scan the images and will look for the required symptoms and visual clues to detect retinopathy. If found positive, the system will find the severity and display the results. Dr. Rao will then verify these results by looking for relevant features in the image.
2. If found negative, the system will show appropriate message.

The system will train on Dr. Rao's verified results which will eventually improve the reliability of the system. In addition to this, Dr. Rao can diagnose more patients in a much more effective way by using this system.

2.2.3 User Interaction

a.)Common errors and mistakes

1. There is a chance that the user may upload incorrect images which will lead to incorrect results from the system.

2.There might be erroneous values provided by the user for some features of the input which may lead to false predictions.

b.)Expected Messages

1.If the image being submitted at the input end is not of retina,the user will be navigated back to the input area with a request message for resubmission.

2.If at the user end,the values being submitted are highly erroneous or out of bounds for a particular feature,the user will be notified with a message specifying inappropriate input instantly.

c.)Ways to retry

2.2.4 Open Issues

- Most screens for input and output will have standard format of uploading images or text files and displaying text result respectively. This document is more concerned with the functionality and the interaction design, not the exact look and layout.

2.2.5 Side Notes

- More than 95% of thyroid cases are autoimmune thyroiditis.
- The symptoms associated with Hashimoto’s thyroiditis are easily misdiagnosed because they are common to many other conditions, including psychological issues
- Diabetic Retinopathy is so prevalent that around 93 million adults are affected by it on a global scale
- Although there’s no absolute cure to Diabetic Retinopathy, diagnosis and treatment during the initial stages help maintain vision.

Technical Note : The software should be used only by doctors to maintain the integrity and accuracy of the system.

2.3 Operating Environment

The software will operate on any computer system with following minimum requirements :
Hardware requirements :

- Intel i3 processor or its equivalent
- 8 GB physical memory
- 8 GB graphics memory

Software requirements :

- Windows 10/ Linux
- Python environment with all required libraries (Keras, TensorFlow, etc.)

2.4 Design and Implementation Constraint

1. For Diabetic Retinopathy :
   a. Data quality : The image quality will determine how precise the system would be. The images are generally over-bright, blurry or are too dark to directly use them. Preprocessing is required for obtaining the correct features from the images.
   b. Variety of data sources : The images will be captured from various sources leading to heterogeneity in image properties. Only a limited types of images can be preprocessed and used in the system.
2. For Hashimoto’s thyroiditis :
   a. Data sources : The data will be obtained from various data sources and preprocessing will be required.
   b. Missing values : The data from certain sources will not have some of the required attributes filled up. Hence the system will have to assume those missing values.
3. General constraints :
   a. Due to average quality of computing systems in user environment, the periodical training of the system will take large amounts of time. To reduce this time, small batches of data would be used to train the system at a time.
   b. The processing unit must support parallel operations as the system will use this property to reduce the training time.
   c. The deployment machine must have python interpreter for running our system.

2.5 User Documentation

1. User Manual : A user manual will be provided to the user to refer to the working of the system if required. A basic training will be given to the user on how to use the system.

The manual will contain all the things described in the basic training and also about the exceptional cases where the system might not work properly. The do's and dont's would also be written in it. The manual will also contain information about how to prepare data before feeding it to the system.

2. Video Tutorials : The user will also be provided video tutorials for learning how to use the system features properly and efficiently. Each feature will be covered in detail in the tutorials. Video tutorials will help new users to get used to the system.

2.6 Assumptions and Dependencies

Software assumptions :

- For reducing time required for training the model, the system should support multiprocessing or should be distributed in nature.
- The python libraries must be up-to-date.

General assumptions and dependencies :

1. For Diabetic Retinopathy :
   a. Both left and right retina images should be available.
   b. Dataset on which the model is trained is universal and authentic.
2. For Hashimoto's thyroiditis :
   a. Data used is from a reliable source and is universal.

3.1 External Interfaces

3.1 User Interfaces

Artificial intelligence is defined as the ability of computer systems to perform complex, independent tasks that require human-like intelligence such as visual processing, speech recognition, or decision-making. Machine learning is employed when computer programs have the ability to improve their own decision making by 'learning' from data provided to them without provided explicit rules. Deep learning, a powerful model of machine learning, utilizes layers upon layers of 'neural networks' to enhance the software's ability to independently perform feature extraction from data.With context to this problem,the user has the ability to input image of the patient's retina for which a user interface is needed.The inputs provided at the user interface end will then be provided as an input to the stated configuration of Neural Network .

3.2 Hardware Interfaces

The hardware in specific for this project will be used as allocated to our Virtual Machine by Google Colab Research.The generic hardware details of the Google Colab Virtual machine are as follows

GPU: 1x Tesla K80 , compute 3.7, having 2496 CUDA cores , 12GB GDDR5 VRAM

CPU: 1x single core hyper threaded Xeon Processors 2.3Ghz i.e(1 core, 2 threads)

RAM: 12.6 GB Available

Disk: 33 GB Available

For every 12 hours the Disk, RAM, VRAM and the CPU cache data allocated to the virtual machine by Google Colab will be erased.In order to tackle the mentioned problem one of the approach is to maintain saver checkpoints during the training of network that allow for restart training from a checkpoint when it is disrupted while other proposed approach is to simply write a script which itself feeds the training data to the model before the resources administered to the virtual machine are deallocated .

3.3 Software Interfaces

Software : Python 3.6

Libraries used :

Tensorflow : r1.10

TensorFlow is an open-source software library for dataflow programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.It will be used for the formulation of the neural networks in problem.

Pandas : v0.23.4

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.Storing and retrieval data as well as the data structures on the basis of which the input will be fed to the neural networks will be developed with the help of this library.

Numpy : 1.15.1

Numpy is a library for the Python programming language which provides support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Operating System : Debian / Windows

Web Browser : Desktop Versions of chrome or firefox

4. System Features

The below features are categorized by mode of operation. For each use case, there are specific requirements which are detailed below.

4.1 Noise Reduction

- The system should provide a noise removal function, which will help in acquiring better and more accurate images.
- Noisy data or outliers should be handled accordingly and appropriately. In case of images, the reduction of noise should be done differently.
- While in case of text, outliers should be detected and removed.
- The system should also provide functionality for filling in missing data, with appropriate approximations for better accuracy.

4.2 Classifier

- The system should provide functionality of classifiers using which the most accurate classified data can be used for detection.
- Classifiers of the system should vary based on the type of input the system receives, i.e., text or image.

4.3 Train System

- The system should provide a login screen for the admin.
- The system should also provide functionality for taking input from the admin to train the model.

4.4 Model System

- An output screen should be provided by the system for the admin to the see the result computed by the system.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Classification time and response time should be as little as possible, because one of the software's features is timesaving.Classification and response times
are very low hence, multiple patient can be diagnosed at the same time.

5.2 Safety Requirements

If there are interruptions like power cut,etc during prediction process then the model can be used again by feeding data after interruptions are removed.

5.3 Security Requirements

The product should be utilised by authorised people of health institute who has access to the data being fed to the model.

5.4 Software Quality Attributes

5.4.1 Usability

The system should be easy to use. The user should get results with basic steps like uploading file and pressing go button. The system also should be user friendly for admins.

5.4.2 Reliability

This software will be developed with machine learning, feature engineering and deep learning techniques. So, in this step there is no certain reliable percentage that is measurable. Also, user provided data will be used to compare with result and measure reliability.
With recent machine learning techniques, user gained data should be enough for reliability if enough data is obtained. The users can reach and use program at any time, so maintenance should not be a big issue.

5.4.3 Supportability

The system should require python knowledge to maintenance. If any problem occurs in deep learning methods, it requires code knowledge and deep learning background to solve.

6. Other Requirements

TBD

Appendix A: Glossary

TBD

Appendix B: Analysis Models

TBD

Appendix C: To Be Determined List

TBD