|  |
| --- |
| **Palmly Neural Network Training:**  Requirements Specification (v 2.0) |

Project: Palmly Neural Network Training

Date: December 4, 2019

Prepared by: Nguyen, Katie; Ruiz, Sofia; Tolliver, Moriah

|  |
| --- |
| **1. Introduction** |

This document contains the system requirements for the neural networks that will ultimately be used in the Palmly mobile application. These requirements have been derived from discussions with Andrew Forney, Ph.D.

**1.1 System Overview**

The Palmly Neural Network Training project aims to train 3 different neural networks to recognize the major hand lines used in palm reading. Each neural network will be trained to recognize a different line on the hand, and will be able to categorize the shape of the line shown on an image from the user. The networks will be trained using TensorFlow, a tool built upon Keras that enables machine learning to be on a device instead of sending data back and forth from a server.

**1.2 Scope of the Product**

The product will be used in a mobile application that will enable users to upload photos of their hands, receive readings based on the perceived shape of their major hand lines, and store and share these photos and readings. The trained networks should be able to perceive the shapes of the major hand lines used with suitable accuracy. However, this project does not include building the application that the networks will ultimately be used in. The accuracy of the networks will be both trained and assessed using TensorFlow.

**1.3 Document Overview**

This document will be organized as follows. Section 2 will detail the requirements for each component of the system.

The following language shall be used to specify requirements:

**1.3.1 “Shall”**

This specifies a mandatory requirement that must be fulfilled by the project.

**1.3.2 “Should”**

This specifies a requirement that may or may not be satisfied by completion of the project.

**1.3.3 “Will”**

This declares a design goal describing the way in which other requirements accomplish a purpose. These goals may or may not be met by completion of this project.

|  |
| --- |
| **2. System Operation and Requirements** |

**2.1 Computer Software Configuration Item Component Breakdown**

The Palmly Neural Network Project will consist of two computer software components (CSCs): the database and the neural networks. The database will hold training data and testing data. The neural network CSC will consist of the Head Line, Heart Line, and Life Line networks.

**2.1.1 Database CSC**

The database will be a Mongo database with a collection of images. The documents in the images collection will be managed and manipulated using the Mongo Shell. The database will be used primarily to monitor participant demographics as data is collected.

**2.1.1.1 Training Data Computer Software Unit (CSU)**

The training data will be stored as image documents database images collection. Each document will have fields detailing the image URL, the label assigned to the image, and the image subject’s demographics.

**2.1.1.2 Testing Data CSU**

The testing data will be stored as image documents database images collection. Each document will have fields detailing the image URL, the label assigned to the image, and the image subject’s demographics.

**2.1.2 Neural Network (NN) CSC**

The neural networks will be trained using TensorFlow Hub. Given an input image, each different network will output a probability for the accuracy of each possible label.

**2.1.2.1 Head Line NN CSU**

This neural network will take an image of a palm as input and output the probable accuracy for the following labels: *short, long/straight, splits,* and *long/curved*.

**2.1.2.2 Heart Line NN CSU**

This neural network will take an image of a palm as input and output the probable accuracy for the following labels: *long/straight, short/straight, long/curved, short/curved, xs,* and *splits*.

**2.1.2.3 Life Line NN CSU**

This neural network will take an image of a palm as input and output the probable accuracy for the following labels: *short, long, faint,* and *curved*.

**3. Functional Requirements by CSC**

The completed system can be expected to have a private, and up-to-date Mongo database holding all the images used for training and testing along with the subject demographics for each image. The system will also have three private neural networks that can be downloaded from Github and used to receive the probable accuracy of each palm-line label on an input image.

**3.1 Database (DB)**

3.1.1 The DB shall have enough capacity for at least 3600 images with their respective labels, URLs, and demographic information.

3.1.2 The DB shall only be accessed by project developers and supervisors.

3.1.3 The DB shall only be hosted on the local machines of project developers and supervisors.

3.1.4 The DB shall have a backup in the form of a csv file with all the up-to-date document information.

3.1.5 The DB shall contain no information that would allow any participant to be identified.

3.1.6 The DB shall consist of one collection named *images*.

3.1.7 The DB shall consist of documents in the *images* collection such that each document has at least an accurate URL and privacy label.

3.1.8 The DB will consist of documents in the *images* collection such that an image may have location, age range, skin tone, head label, heart label, life label and reading label attributes.

3.1.9 The DB shall allow for new fields to be created for individual documents as needs arise.

3.1.10 The DB shall have a shell in which queries can be executed.

**3.2 Neural Networks (NN)**

3.2.1 The NN shall have a command line interface.

3.2.2 The NN shall have a command for receiving an input image.

3.2.3 The NN command line interface shall have a command for displaying accuracy graphs.

3.2.4 The NN shall be trained on data that has been labeled by the designated label for the corresponding line.

3.2.5 The NN output files shall be able to take images of file extensions *.jpeg*, *.jpeg*, or *.png* as input.

3.2.6 The NN shall output a different accuracy probability for each possible label.

3.2.7 The NN shall output each probability as a floating-point number.

3.2.8 The NN shall output probabilities with at least 4 significant figures.

**4. Performance Requirements**

**4.1 Results should have at least 60% accuracy**

The accuracy reported by TensorFlow Hub should be at or above 60% when the NN is trained on 900 or more pictures.

**4.2 The size of the combined training and testing data shall not exceed 15MB**

The size of all the images gathered during data collection must be within 15MB in order to be stored on Google Drive without a fee.

**4.3 Each label used in training must correspond to at least 100 images**

If any label corresponds to less than 100 images, it should be removed from the set of labels for that hand line during training.

**5. Project Environment Requirements**

**5.1 Development Environment Requirements**

5.1.1 The DB and contained data shall only be accessible to the project developers and supervisors

5.1.2 The NN shall only be accessible from the private Palmly Github repository.

5.1.3 The NN will only be trained on the personal machines of the project developers.

5.1.4 The personal machines of each developer shall have enough space to hold the training and testing data for their assigned palm line.

5.1.5 The personal machines of each developer shall be compatible with Python2 and Python3.

5.1.6 The personal machines of each developer shall be compatible with the latest version of TensorFlow Hub as well as one version prior in case of deprecation.

5.1.5 The personal machines of each developer shall be compatible with Python2 and Python3.

**5.2 Execution Environment Requirements**

5.2.1 The NN shall only be executed on the personal machines of the project developers and supervisors.

5.2.2 The users shall not need to have a copy of the data to run the NN.

5.2.3 The users’ machine will have enough space to hold the classifier models for the NN they wish to use.

5.2.4 The users’ machine shall be compatible with cdPython2 and Python3.

5.2.5 The users’ machines shall be compatible with the latest version of TensorFlow Hub as well as one version prior.