|  |
| --- |
| **Palmly Neural Network Training:**  Software Design Description (v 1.0) |

Project: Palmly Neural Network Training

Date: 6 November 2019

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

|  |
| --- |
| **6.1 Introduction** |

This Software Design Description contains high-level information regarding the training of the neural networks that will ultimately be used when designing the Palmly mobile application. The team envisions Palmly to be an app that provides a user with the opportunity to “scan” their left palm by uploading an image of their hand. The picture will then be analyzed by the aforementioned neural networks. After a user has finished uploading their palm, the user will be presented with their own specific palm reading. The team intends to add a lifestyle component to the app, once the palm reading portion has been completed. The lifestyle component will be individualized based on the user’s reading. The final product of this project will be the mobile application that users can download and use, but the primary focus of the team for the first part of this project will be the training of the machine learning agent. The training for the agent will be implemented through November and December 2019.

**6.1.1 System Objectives**

The objective of this project is to train three different neural networks to then create an app that a user could use to obtain a palm reading. Three neural networks will be trained such that each one recognizes one of the three main lines on a user’s left palm: the heart line, head line, and the life line. Twelve hundred labeled images will be used to train each of the neural networks so that each one focuses specifically on one palm line. Once the three neural networks have been trained, a front-end to the app will be made so that users can begin to upload their own images and receive a reading of their palm. In addition, a lifestyle component to the app will be added once the front-end is being incorporated, which will be individualized based on each user’s palm reading.

**6.1.2 Hardware, Software and Human Interfaces**

**6.1.2.1 Team Member Personal Computers and Devices**

To collect data, every team member may use their personal laptop or mobile device to capture images of left palms to cultivate the data set. The only requirements for these devices is that they are able to send, receive and capture images of types jpeg or png and that they can connect to Google Drive. The team members will also use their respective devices to test the neural networks throughout development and testing.

**6.1.2.2 Computer for Data Storage, Training, and Testing**

An Intel NUC7 with an i7 processor will be used to store the sorted data sets and to train the neural networks using TensorFlow.

Max Memory Size: 32 GB

Max Memory Bandwidth: 34.1 GB/s

**6.1.2.3 Google Drive**

This application automatically updates to the latest version. This web application will be used to collect, store, update, and manipulate data within the dataset. Drive itself will be used to store the data. Scripts will be used to access the data, label the images, and insert the image information into Sheets. Sheets will be used to convert the data into a csv so that the data may be transferred to MongoDB. Forms will be used to collect new data from volunteers. Slides will be used to present the findings and test results of the project.

**6.1.2.4 Atom: Text Editor (v1.40.1)**

This application will be used for the implementation and testing of the neural networks using TensorFlow.

**6.1.2.5 Chrome (v66.0.3359.181)**

We will be using v66.0.3359.181 of the application to access the project’s Google Drive as well as the documentation for TensorFlow.

**6.1.2.6 TensorFlow: Machine Learning Platform**

We will be using v2.0 of this platform to train and test each neural network.

**6.1.2.7 Python**

We will be using version 3 of the Python language to interface with TensorFlow. Python will be running on the terminal of the Intel NUC7

|  |
| --- |
| **6.2 Architectural Design** |

Throughout the duration of the Palmy Neural Network Training Project, the design of the system will resemble a data flow architecture. The first class consists of a Data Storage Control. In this sense, the palm image data will be collectively organized and maintained in a shared Google Drive across all team members. Additionally, the images, along with the hand’s reading and other demographic information, will be stored in a MongoDB database so that queries can be made easily to inquire specific information in regards to the demographics and diversity of the dataset. From here, the data will be piped into a Sorting Interface Control that allows for the renaming and sorting of images based on the line categorization that is given to them. Finally, the data will be locally downloaded in order to be streamed into the TensorFlow Serving for machine learning production environment. It is within this TensorFlow Servable that the data will be utilized in order to retrain top layers of neural networks through transfer learning.

**6.2.1 Major Software Components**

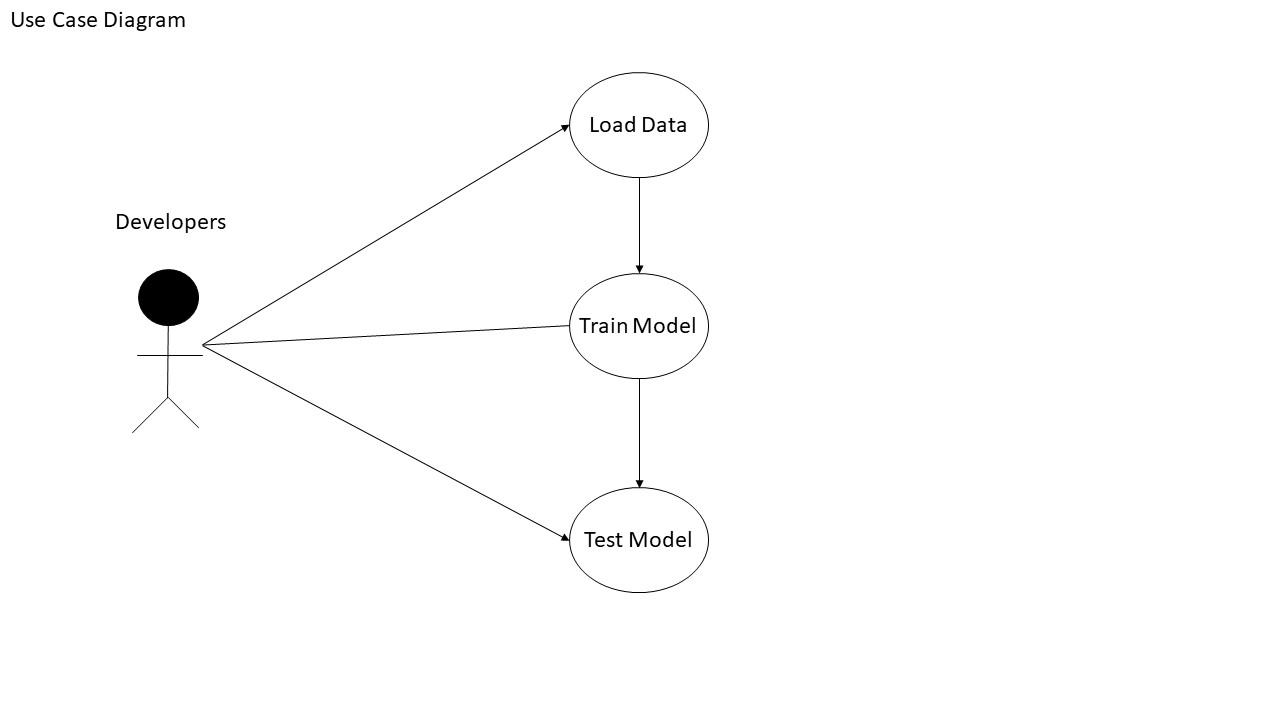
The functional requirements that have been outlined in the Palmy Neural Network Training Project consist of the ability to load and store images of left palms in a shared platform that is accessible to all team members. Additionally, with this organized collection of images, this project requires that the dataset can be fed into a training pipeline that will result in three separately trained neural networks, one for each of the major hand lines.

There are multiple software subsystems that work in relation to one another in order to be able to successfully carry out all functional requirements. For instance, the team has a Google Drive and MongoDB setup to store the actual images as well as all demographic info pertaining to said data. The Google Drive, along with a sorting interface, allows for the renaming and relocating of images. In this sense, the images can be grouped with all other images that have been given the same hand line classification. Finally, the use of an open source library to help train ML models, TensorFlow, provides the API calls that allow for the retraining of models in order to produce neural networks capable of classifying new palm images.

**6.2.2 Major Software Interactions**

In regards to the software interactions of all components in the project, there is a clear parsing of data between both database implementations and the sorting interface. For example, the sorting app is developed through Google’s scripting language that interacts directly with the data being stored in our Drive. In the code written for the sorting app, each image in the Drive is iterated through and allows for the user to select a button describing the hand line that is being observed. Then, a copy of the image is stored in a different directory within the Drive based on the certain hand line attribute that was given to the picture. This allows for all for all of the data to be sorted into either the Life, Heart, or Head line folder with only the label that pertains to that specific line. Once all of the data has been sorted, it is downloaded to a local machine so that the folder can then directly be fed into the training pipeline. In this sense, TensorFlow is not directly communicating with the database itself, but the necessary pieces of the database can be downloaded and used in conjunction with TensorFlow when needed.

**6.2.3 Architectural Design Diagrams**

****

