McMaster University

SFWRENG 4G06

Tip of My Shoe - Development Process & Implementation

Member Name	Mac ID	Student Number
Chris DiBussolo	dibussoc	400070368
Andrew Lucentini	lucenta	001430150
Owen McNeil	mcneilo	400065750
Daniel Scime	scimed1	400069926
Ashley Williams	willia18	400081787
Lucas Zacharewicz	zacharel	400054446

November 18, 2019

Contents

List of Figures

List of Tables

1 Revisions

Revision Number	Date	Comments
0	November 18, 2019	Initial document.

Table 1: Table Of Revisions

2 Roles And Responsibilities

2.1 Chris DiBussolo

- Documentation
- Algorithmic Analysis and Design (Relative scoring etc.)
- Data-set labelling
- Debugging

2.2 Andrew Lucentini

- Documentation
- Back end - SQL, PHP

2.3 Owen McNeil

- Documentation
- Back end - API, Database / Storage
- Back end - Server Provisioning

2.4 Daniel Scime

- Documentation
- Front end - UI/UX
- Consistency and continuity throughout code and documents
- Style and appearance

2.5 Ashley Williams

- Documentation
- Front end - UI/UX
- Data-set labelling

2.6 Lucas Zacharewicz

- Image detection
- AI model training
- Web scraping for shoe images

3 Version Control

Git will be used to keep track of all our documentation and code.

4 List Of Development Tools

- Python 3
- Selenium and LabelImg for web scraping
- ImageAI and TensorFlow for machine learning and image recognition

- HTML, CSS, JavaScript for front-end
- MongoDB, Node.js, Express.js, Nginx for back end

5 Process Workflow

5.1 Step 1: Building a Data Set for Image Recognition

Inputs: Web scraped photos of shoes

Outputs: Data set to be used for model training

Acceptance Criteria:

Data set must contain at least 25 distinct shoes and each shoe should have at least 200 photos.

5.2 Step 2: Building a Basic AI Model

Inputs: Previously built data set

Outputs: Trained model for the data set

Acceptance Criteria:

A photo of a shoe that exists in the data set should be detected with at least 80% confidence.

5.3 Step 3: Front End website/application design

Inputs: N/A

Outputs: Intuitive interface

Acceptance Criteria:

A user can supply images and appropriate results will be returned. Users should be able to navigate the interface with little to no instruction.

5.4 Step 4: Back End Development and Website Algorithm Design

Inputs:

- Image detection / AI scripts

- Shoe images

Outputs:

- Exposed API
- Likeness Ratings

Acceptance Criteria:

The API exposes an endpoint that accepts a user's image and sends it to the image detection scripts. The endpoint should return an object containing potential matches, their confidence levels, and any meta data such as shop links.

5.5 Step 5: Testing and Debugging

Inputs: Test cases (unexpected input, error handling, etc.)

Outputs: Repeatable test suites

Acceptance Criteria: The program must pass all of the test cases.

6 Details On Steps To Be Taken

6.1 Step 1: Building a Data Set for Image Recognition

Tools To Be Used

- Python 3
- Selenium
- LabelImg

Special Instructions For Tools

- In particular, **Selenium**, an open source web automation tool will be used to scrape the web for images of desired shoes to build an initial data set to be used for image recognition AI training.

- LabelImg used to label images in the data set with yolov3 format to create relevant .xml files that our model uses to train

What To Put Under Version Control

- Python code for the web-scraping. After initial use for testing, can be re-used for building larger data sets.
- The data set. (Probably better to put it in a google drive as it is expected to grow beyond the allowable size of github)

Contributors

- Web-scraping: Lucas Zacharewicz
- Image Labelling/data-set building: Project Team

6.2 Step 2: Building a Basic AI Model

Tools To Be Used

- Python 3
- ImageAI
- TensorFlow (v 1.15, includes gpu accleration)

Special Instructions For Tools

- Need to have a version of TensorFlow < 2.0 as imageAi doesn't support > 2. Consequently, Python 3.7 or lower is needed as TensorFlow 1.15 only works on versions of Python up to 3.7

What To Put Under Version Control

- Generated models, only the highest epoch from each batch (lower epochs saves of the model are generally less accurate)

Contributors

- Lucas Zacharewicz - Model training

6.3 Step 3: Front End website/application design

Tools To Be Used

- HTML, CSS, JavaScript

Special Instructions For Tools

- N/A

What To Put Under Version Control

- All the code files required for the website to run.

Contributors

- Daniel Scime, Ashley Williams

6.4 Step 4: Back End Development and Website Algorithm Design

Tools To Be Used

- MongoDB, Node.js, Express.js, Nginx, PHP, SQL, Python3

Special Instructions For Tools

- Python3 will be used for consistency with the machine learning code. The main algorithms/functions to be developed are ones dealing with the likeness calculation that gives a quantifiable value to how well the matched shoe resembles the target image.

What To Put Under Version Control

- All the code files required for the website to run.

Contributors

- Owen McNeil, Andrew Lucentini, Chris DiBussolo

6.5 Step 5: Testing and Debugging

Tools To Be Used

- TBD

Special Instructions For Tools

- TBD

What To Put Under Version Control

- All the code files required for the website to run.

Contributors

- Owen McNeil, Andrew Lucentini, Chris DiBussolo

7 Handling Changes

Github's issue tracking will be used to deal with changes to development artefacts. All issues will be labeled appropriately according to their "type" (e.g. bug, enhancement, etc). Functional changes will be prioritized over non-functional. Issues will be assigned to members as needed and should be related to their delegated role. During weekly group meetings, we will discuss issue resolution, verify each other's work and determine other possible changes.