Design Document for ULearn: helping frustrated students with web-based learning

Authors: Lynne Grewe
Date: June 12, 2018

Status: approved

Concept Summary
The goal of this work is to provide feedback to students regarding their engagement/emotion/frustration while utilizing web-based instructional material using computer vision and machine learning techniques. An Android Application will be developed that will present a simple user interface of a web-browser to the student. Based on computer vision machine learning the students facial expressions are interpreted. If the student seems to be experiencing frustration or anger the user is presented with a pop up giving different suggestions. Using the corresponding webpage’s title and meta description tags a set of search tips related to currently Viewed webpage will be presented. Version 2: Google NPL Text analysis or other web services will be explored for use in tips creation.

Audience/Customer
The initial test audience will be students at CSUEB in Computer Science. Once user/acceptance test is completed the target audience will be any students who use web-based material. The application has the potential of working with anyone, not just students, who view web-based materials and could be assisted when “frustrated”.

Background
The application will be developed as an Android mobile application. Some of the core technologies involved in this project including Computer Vision, Machine Learning and Text to Speech. The front facing camera on the android device will be used to capture images of the user as they are using the application. Each
image or periodically sampled images will be processed through a Machine Learning using Convolutional Neural Networks to detect the user’s emotions.

**Application Cost and Projected success** (optional)

Version 1 application will run completely on the phone and will not incur any costs. Version 2 that explores the use of NLP services will incur costs and a budget will need to be invoked and once exceeded only freely generated tips version used.

Success is highly dependent on deployment to playstore once acceptance testing is done. No current plans for marketing or advertising.

**Interface Mockups**

Diagnostic interface will show camera
Design - Use Case Diagram(s)

Following the use case diagram above the following shows a more illustrative system diagram

- **Camera Capture Module**: Control and capture of image frames
• **Emotion Detection Module** = using Machine learning
  ○ Version 1.A = using Tensorflow CNN
  ○ Version 1.B = using Google cloud vision
  ○ **Face Detection** = because the project is less than 2 weeks to create for version 1, and student is “close” to device to be able to read webpage we are going to test what results we can get without a prior face detection stage. For version 2 we will look at adding face detection leveraging OpenCV.

• **Generate Tips Module** = initial version using currently viewed webpage title and description. Google Search will be used in the creation of tips, which are links to (hopefully) related web pages. Potentially, a future version, can extend to use semantic Natural Language Understanding to generate further search tips.

**Related Work**


**Frameworks/Services/Cloud/Backends**

The following are to be used in ULearn

• Android = mobile platform
• Tensorflow = for CNN development
• Google Cloud Vision Face Detection service = to compare with Tensorflow and in case schedule is tight and can not complete Tensorflow. Disadvantage over using CNN on mobile device is that there is a cost involved. The requirement of being on the Internet is already necessary for the application as the student is looking at web-based material in ULearn already.
• OpenCV = Computer vision framework with support in Android, used for image manipulation.

**Testing**
Do to the quick (<2 week ) timeline, unit testing will be the sole responsibility of each developer on the team. Integration testing will be performed during Code Review time.

**Schedule**

Will use Trello board for monitoring Schedule. The following snapshot of the schedule is shown below and is already in progress.

**Dependencies**

- Access to data and tensorflow model for retraining is necessary but this is already obtained.