SANTA CLARA UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

Date: December 1, 2017

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Manoj Adhikari
Colby Harper
Sean Karstein

ENTITLED

On the Construction of Matter, or Is There a God Particle?

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Thesis Advisor

Thesis Advisor

Department Chair

Department Chair

On the Construction of Matter, or Is There a God Particle?

by

Manoj Adhikari
Colby Harper
Sean Karstein

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering
School of Engineering
Santa Clara University

Santa Clara, California
December 1, 2017

On the Construction of Matter, or Is There a God Particle?

Manoj Adhikari
Colby Harper
Sean Karstein

Department of Computer Engineering
Santa Clara University
December 1, 2017

ABSTRACT

Taking group photos during important events is a common practice. Group photos are taken to remember cheerful times when people had opportunity to meet many other people. However, an unappealing facial experience of one person can easily ruin the entire photo. Capturing the wrong moments when a person doesn't look attractive can leave him/her displeased from the complete event experience. A solution is to develop a mobile app that captures the moment when everyone is smiling with eyes wide open. Our solution aims to develop an iPhone app that will preclude users from worrying about not having a great group picture.

Table of Contents

1 Introduction **1**

2 Requirements **2**

2.1 Functional Requirements 2

2.1.1 Critical: 2

2.1.2 Recommended: 2

2.1.3 Suggested: 2

2.2 Non-Functional Requirements 2

2.2.1 Recommended: 2

2.2.2 Suggested: 3

2.3 Design Constraint 3

3 Technologies Used **4**

4 Design Rationale **5**

5 Test Plan **6**

6 Risk Analysis **7**

7 Ethical Analysis **8**

List of Figures

Chapter 1

Introduction

Taking pictures to save our best memories is a common practice. During group pictures, photos can be entirely ruined by a single person?s expressions. We all have seen photos of people who did not smile/looked away, or worse, been that person. The experiences at big events and gatherings would be much more fulfilling if we didn?t have to take many pictures just to get a nice smiling photo. Modern cameras and phones have beautiful design incorporated with advanced features. However, they have forgotten to address this pesky issue that would uplift everyone?s picture taking moment.

Currently, the only group photo technology to be created is the automatic photo with timer. The purpose of this is to eliminate the need for someone to click the button to take the photo, however it is not smart. It does not signal when the photo will be taken, and takes the photo only when the timer has reached zero. Therefore, this does not have any effect on the quality of the photo. Currently, the only solutions affecting the quality of group photos are after effect softwares such as photoshop. However, these often require a large amount of money, and a level of skill in using the product, which most people don?t have.

We propose to build an iOS app that utilizes the iPhone?s camera and adds the feature of taking pictures automatically when everyone in the group is looking at the camera with their eyes open and smiling. The app will automate the tedious task of capturing the right moment saving people?s time for actually enjoying the moment. Computer vision will be used to narrow the focus of the image, to then allow a machine learning model to determine when the right moment to snap the photo is.

To conclude, our design will accomplish two things: firstly, it will save time for both people who are getting their photo taken as well as the person taking the photo, as the camera will know exactly when to take the right photo, and secondly, will improve the quality of photos, allowing for memories to be better cherished by all.

Chapter 2

Requirements

2.1 Functional Requirements

2.1.1 Critical:

- Users can take pictures using iPhone camera.
- Users can choose to save or disregard the photo captured.
- The app can identify if everyone in the group is smiling with eyes wide open or not.
- Photos are automatically captured without users click.

2.1.2 Recommended:

- Users can share pictures with their friends using phone numbers.

2.1.3 Suggested:

- Users can give feedback on whether the app captured good pictures.
- The system will offer photo editing features

2.2 Non-Functional Requirements

2.2.1 Critical:

- The user interface will be simple and intuitive
- The system will be easy to install and use.

2.2.2 Recommended:

- The system will be secure from unauthorized users.

2.2.3 Suggested:

- The system will be high performance and will require minimal system resources.

2.3 Design Constraint

- The app can only run on iPhones with iOS versions starting iOS 10.3.

Chapter 3

Technologies Used

- Hardware
  - Development
    - Macbook Pro
    - ThinkPad
  - Application Testing
    - Iphone 7
    - ThinkPad with Iphone emulator
- Programming Languages
  - Swift
    - For iOS Programming
  - Python
    - For Machine Learning Programming
- IDEs
  - XCode
  - MacVim (or the like)
- APIs
  - TensorFlow

Chapter 4

Design Rationale

We are using a data flow model to increase the speed at which the application will operate. We want the device to recognize when the right moment to take a picture is as fast as possible so that the perfect photo is taken. Because of this, we decided not to have any communication with a server, and instead keep all logic on the local device.

We are using Iphones and the associated iOS devices and software for testing and development because that is what most of our team uses, as well as most of the people we know at Santa Clara University. This will allow for a potentially large alpha testing process during the later stages, as well as eliminating the need to buy iOS phones for beta testing.

Making use of the existing API TensorFlow will allow us to avoid rewriting existing code, saving a lot of time on development.

We will develop the application in XCode with Swift because they are the designated language and editor combination with iOS to make applications. We will use Python to design the machine learning component as it is well documented in coordination with TensorFlow.

Chapter 5

Test Plan

Unit testing will be done throughout the quarter from Week 1-10 on every new functionality we implement. We plan to perform a lot of unit testing from week 1-5. Unit testing is suitable during this period because we will be heavily invested in implementing the individual functions that make our system. From week 5-7, we combine smaller functions into a full-scale working application. We will perform a lot of black-box testing and system testing to evaluate our product. From week 7-10, we will scale up the amount of our test data and perform stress testing. We plan to use group photos datasets to test the smiley-face recognition feature of our system. Once we combine that feature with iPhones photo capturing feature, we will take photos using our iPhones to verify that our system works.

Chapter 6

Risk Analysis

Risk	Consequences	Probability	Severity	Impact	Mitigation Strategies
Illness	Portions of development blocked.	.4	5	2	Wash hands and get flu shots.
Insufficient Development Knowledge	The system will no accomplish our goals.	.2	7	1.4	Use online resources and communicate with team members when a roadblock is hit.
Coordination Failure	Components overlooked and unfinished	.15	8	1.2	Keep an organized schedule of due dates and follow the development timeline.
File loss	We lose access to our files.	.05	10	.5	Use GitHub to protect files, and always push to master when updating.

Chapter 7

Ethical Analysis

Before developing our product, there are several ethical scenarios that need to be considered and addressed. First, we must acknowledge internal ethical dilemmas that we may encounter. The first of which is ensuring that every team member has a voice that is heard during the design process. No member of our team should be ignored or blocked from the development process under any circumstances. On the opposite end, no member should be forced to perform a greater amount of work than the rest of the group. The workload should be distributed among team members as evenly as possible. To prevent either of these ethical issues during the design, one solution is to document every necessary next steps as well as all recently completed steps when all members are present, or on an online forum that all members have access to. This will allow all members to be aware of and able to add input to every task, preventing anyone from being blocked from development information. And this will allow all members to see who has done which tasks, so it will be easy to determine if the work load is being divided fairly.

Secondly we must acknowledge external ethical dilemmas that could be pertinent to our product. One of which, is ensuring that our product works for people of all races and genders. We do not want a product that is implicitly racist or sexist due to its code. In order to ensure this is not the case, we plan on having diverse data that incorporates people of all races and genders, as well as test cases that confirm all people will be able to share the same experience while using our product.