**Project Identity**



Miguel Saldana

Project Success

Project Identity is a facial recognition engine that will detect faces in pictures. Through Project Identity, I would like to create an Application Programming Interface (API) that will allow developers to use my facial recognition implementation, which will allow them build upon and create their own facial recognition applications. Project Identity will allow developers to identify faces in pictures, as well as being able remember a specific face. A stretch goal that I have in mind for Project Identity is the ability for the engine to learn and adapt to different features that can be present on a face. I will create Project Identity from scratch, without the use of any third-party libraries. Project Identity is a major component toward a future project I would like to complete, which involves facial recognition.

Target Users

Developers interested in facial recognition will also be able to make use of Project Identity. Developers interested in facial recognition will be able to use the API for their own facial recognition engine simply by using my implementation. By allowing developers to use Project Identity, they will not have to implement their own facial recognition engines. The Project Identity API will allow developers to identify, remember, and search for faces across several pictures.

Project Backlog

1. “As a software developer, I want to implement the Project Identity API without having to create my own facial recognition engine. Creating a facial recognition engine is difficult and time consuming.”

2. “As a software developer, I don’t want the software to return false matches for a face. All those wrong matches take more time to sort through than they’re worth.”

3. “As a software developer, I want the software to be efficient and return positive matches for a face. I don’t want wait for the matching, I want it to be as fast as possible.”

4. “As a software developer, I do not want an inanimate object to be returned as a face. I expect the facial recognition engine I implement to be able to differentiate between people and objects.”

5. “As a software developer, I want the API to be easy to use, no unnecessary methods, clean and simple to learn.”

6. “As a software developer, I want the API to allow to me specify what type of return results I would like to receive. I do not want the API to restrict me in terms of what it allows me to work with.”

7. “As a software developer, I want the API to have several ways of returning results, the API should allow results to come back as a list of pictures, the pictures themselves, or the inputted pictures with boxes surrounding the face.”

8. “As a software developer, I want a easy to implement way of remembering a specific face in a picture and being able to not only that face but other faces a separate entities for future use.”

2-week Plan

|  |  |
| --- | --- |
| **1st Week** | **Day Plan** |
| Monday | Research bitmaps in-depth, evaluate the effectiveness of comparing two bitmaps  Research how bitmap comparison should be evaluated  Research API structure, requirements, and deployment |
| Tuesday | Research if it’s possible for bitmaps of environments can be mistaken for bitmaps of faces  Research and conclude what percentage should be allowed to recognize something as a face |
| Wednesday | Research the general features of a face  Create general face bitmap  Compare likeness to general face example |
| Thursday | Create bitmaps of general face in several angles  Research how size differences between a bitmap and an image-turned bitmap affect comparisons  Research algorithm that checks regions of an image for face |
| Friday | Research the effectiveness of identifying faces in a colored image versus a black and white image  Research algorithm that converts images into bitmaps |

|  |  |
| --- | --- |
| **2nd Week** | **Day Plan** |
| Monday | Build algorithm that turns images into bitmaps  Build algorithm that changes image size  Take sample pictures of a face to use in comparisons  Research algorithm that compares bitmaps |
| Tuesday | Have image-to-bitmap algorithm completed  Have image size manipulation algorithm completed  Conclude research on bitmap comparisons  Begin outline of bitmap comparison algorithm |
| Wednesday | Test image-to-bitmap algorithm with sample face pictures by creating a bitmap  Build algorithm that compares bitmaps and returns percentage |
| Thursday | Test algorithm bitmap comparison algorithm  Determine if comparison percentage is acceptable |
| Friday | If percentage is not acceptable, then improve algorithm to meet criteria  If percentage is acceptable then find pictures of faces and environments to test algorithm on  Begin research on methods that would isolate the face from the surrounding environment in picture  Begin formatting code to meet API standards |