Android App for Enriching Museum Visits
Project Proposal

Seung Yeon (Stephany) Rhee

seungyeon.rhee@yale.edu

Yale University
Spring 2016

Abstract

This project will consist of making an Android app that provides fun, useful information on art pieces to museum visitors. The app will enrich the experience of visiting a museum for regular visitors, and make the experience more inviting for those who would normally not visit a museum. The motivation for this app lies in the mismatch between the plethora of professional analysis that exist on artworks and the limited experience that museumgoers have today. This mismatch prevails due to the inconvenience of having to search for an artwork on Google, decide which search result to click on, and then having to parse long articles of text to find only a few sentences that are relevant to the piece. This Android app aims to bring the useful information on artworks closer to the user as well as promote discussion on artworks among museumgoers.

Product and Technology

A user might open up this app if he is confused by a piece or is so intrigued by it that he would like to know more about it. He tells the app what piece he is looking at by using basic search, optical character recognition, or image recognition (see descriptions under Detecting Artwork). The app then provides fun and useful information on the piece using curated professional analysis and/or comments from other museumgoers (see descriptions under Providing Useful Information).

Detecting Artwork

There are three methods for detecting the piece in question, listed below in order of increasing technical difficulty. I plan on first implementing basic search, then optionally the other two methods depending on my overall progress in the app at that time.

- **Basic search** requires the user to input the artist name and the piece title.
- **Optical character recognition** (OCR) requires the user to take a photo of the museum label and use an OCR API to read the text for the artist name and the piece title. Options for OCR API include Ocrad.js\(^1\) and Google’s OCR API\(^2\).

---

\(^1\) [http://antimatter15.com/ocrad.js/demo.html](http://antimatter15.com/ocrad.js/demo.html)

\(^2\) [https://opensource.com/life/15/9/open-source-extract-text-images](https://opensource.com/life/15/9/open-source-extract-text-images)
• *Image recognition* requires the user to take a photo of the piece and use an image recognition API to detect the piece. Options include Recognize.im\(^3\), Google Cloud Vision API\(^4\), CloudSight\(^5\), Imagga\(^6\), Moodstocks\(^7\), LTU Technologies\(^8\), and Clarifai\(^9\).

**Providing Useful Information**

What kind of information on the piece would the user like to see? A basic solution would be to display the first paragraph of the relevant Wikipedia article like Google Search does. However, we want the information to be useful to our user; displaying Wikipedia content does not provide any additional value than doing a simple Google search. Therefore, I will implement one or both of the following solutions to provide information that is useful and fun:

• Show a note written by the museum. This would require asking the Yale University Art Gallery to provide notes for their current exhibit, and optionally, an interface for inputting the notes if this is to be scaled to other museums.
• Show comments on the piece by other visitors.

I will use Firebase, which is a BaaS (backend as a service) for data storage management.

**Social**

For now, I’d like to refrain from adding a social aspect to the app; the app will not require the user to make an account or connect to Facebook, etc. This means comments written by visitors will have to be anonymous.

**Testing**

I will test my own app at the Yale University Art Gallery. The app will be tested for quality in the following buckets:

1) Intuitive design.
2) Performance (speed). We don’t want a slow app that is reminiscent of the inconvenience in searching for a piece on Google, clicking on an article, reading long text, etc.
3) Accuracy of *optical character recognition* and *image recognition*, if implemented.

**Adoption**

Upon near completion of the app, I would like to approach the Yale University Art Gallery and ask if they can promote the app to their visitors.

---

\(^3\) [https://www.recognize.im/](https://www.recognize.im/)
\(^4\) [https://cloud.google.com/vision/](https://cloud.google.com/vision/)
\(^6\) [https://imagga.com/](https://imagga.com/)
\(^7\) [https://moodstocks.com/](https://moodstocks.com/)
\(^8\) [http://www.ltutech.com/](http://www.ltutech.com/)
\(^9\) [http://www.clarifai.com/#demo](http://www.clarifai.com/#demo)
Timeline and Deliverables

Deliverables include an Android app and a written final report at the end of the semester. They will be completed according to the following week-by-week timeline.

February 14th  Skeletal Android app (activities, resources, etc)
February 21st  Frontend components of basic search (search bar, buttons, etc)
               Backend of basic search (connect to Google Search API)
February 28th  Frontend of artwork’s information page (artist, title, year, etc)
               Frontend of comment page
               Backend of comments (connect to Firebase; implement storing/fetching)
March 6th      Test, fix, and finalize features of the minimum viable product
March 27th     Research and develop optical character recognition
April 3rd      Research and develop image recognition
April 10th     Test, fix, and finalize the improvement features
April 17th     Test, fix, and finalize the improvement features
April 24th     Write final report