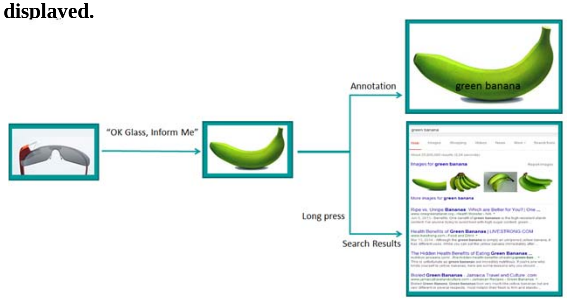
RUTGERS
WINLAB | Wireless Information Network Laboratory

Mini Jarvis – Live Annotations with Google Glass

Saumya Soman, Jin Bing Lin, Anthony Yang, Yash Sanghvi
{saumi89, jinnybinglin, yanga23, ysanghavi1}@gmail.com

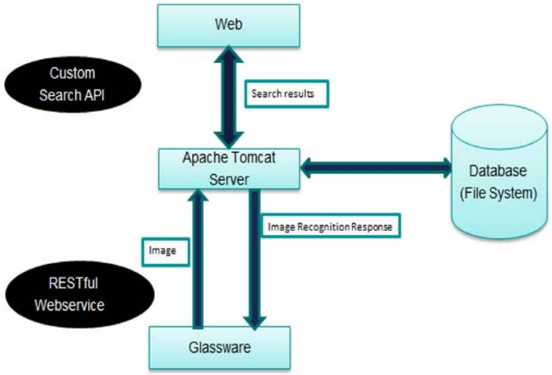
Introduction

Our project Mini Jarvis, in reference to Marvel's Iron Man, is an easy to use, prototype glassware which identifies a particular face or object in the camera preview and provides live annotation. Furthermore, with a long press on the touchpad, Google search of the selected annotation is displayed.



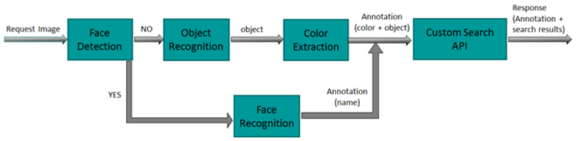
Architecture

- ❖ **Glassware takes images every 10 seconds.**
- ❖ **Image is sent to the server by RESTful web service.**
- ❖ **Server does face/object detection and recognition using face/object trained XML files and color extraction to formulate the annotation required for google search**
- ❖ **Server retrieves search results using Google Custom Search API**
- ❖ **Image recognition response is formed by the annotation and search results**
- ❖ **Annotation is displayed on the glassware.**
- ❖ **Long press by the user yields search results.**



Inside the Server

- ❖ **Face detection is done by OpenCV haarcascade classifiers.**
- ❖ **For face/object recognition the system must be trained and the face/object should be in the database (file system). The identified face/object is enclosed within a rectangle.**
- ❖ **For color extraction, the image within the rectangle is rescaled to 32X32. The most dominant color is found by iterating through the pixels of the rescaled image.**
- ❖ **Including the color of the object along with the object identified returns more relevant search results.**



Training

- ❖ **Training requires two types of samples: positive and negative.**
  - **Positive- images that contains the object.**
  - **Negative- images which does not contain the object.**
- ❖ **For training-: 1500 positive and 600 negative samples.**
- ❖ **Training done by *train_cascade* utility of OpenCV which creates an XML that consolidates information from all the 2100 images. Positive samples are created by *create_samples* utility. Negative samples has to be created manually.**
- ❖ **More stages of training yields more accurate results. Having 20-25 stages of training is recommended. Higher stages take longer to train.**
- ❖ **We trained for one object, banana, and one person, Anthony.**

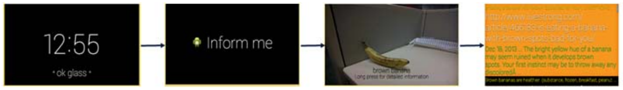
Challenges Faced

- ❖ **Training takes several days to complete**
- ❖ **Accuracy of the training and color extraction**
- ❖ **Over-heating of the Glass**
- ❖ **Delay in response from server**
- ❖ **Glass has frequent updates which can cause code to fail**

Softwares Used

- ❖ **Glassware is built using Android SDK add-on Glass Development Kit (GDK).**
- ❖ **Server side code is developed in Java technology using Eclipse Juno IDE.**
- ❖ **RESTful webservice is in Java with the JAX-RS reference implementation Jersey. REST (Representational State Transfer) is an architectural style which uses HTTP protocol for the CRUD operations (GET, POST, PUT and DELETE).**
- ❖ **Web service has JSON request and response and is deployed in Apache Tomcat 7, an open source software implementation of the Java Servlet and JavaServer Pages technologies.**
- ❖ **OpenCV Library or Open Source Computer Vision is a software library for computer vision. It is cross-platform and mainly written in C++. It has interfaces for Java.**
- ❖ **Google Custom Search API returns JSON response of search results for the annotation. The API allows 100 free requests per day. The response depends on the query parameters.**

Screenshots



Dec 18, 2013 ... The bright yellow hue of a banana may seem ruined when it develops brown spots. Your first instinct may be to throw away any discolored...

Future Work

- ❖ **Multiple simultaneous annotations.**
- ❖ **Optimize the request-response time from server.**
- ❖ **Reverse image search along with annotation.**
- ❖ **Train more objects/faces and with better accuracy.**

References

- ❖ **https://developers.google.com/custom-search/json-api/v1/overview**
- ❖ **https://developers.google.com/glass/develop/**
- ❖ **http://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html**

Acknowledgments

Special thanks to Ivan Seskar, Roy Yates and Shridatt (James) Sugrim for their guidance.

Source code available at : https://github.com/SaumyaSoman/MiniJarvis
Further information, visit https://sites.google.com/site/minijarviswinlab/

WINLAB