**COSC 189.02 – Lab 1 – Team # XX**

Sanket S. Joshi ([f0042hj@dartmouth.edu](mailto:f0042hj@dartmouth.edu)),Anmol Chachra ([f0041wf@dartmouth.edu](mailto:f0041wf@dartmouth.edu))

**Abstract**

This project aims to visually classify 5 fruits, namely apple, guava, peach, mango and pomegranate. In order to be able to recognize the most general image to a good extent, the assortment of images includes 800+ images of fruits that vary in background, multiplicity, cuts, levels of ripening, exposure, etc. These are then split into training and test sets. The python code is a web application using Flask API on top of the Watson visual recognition API that extends the classifier functionality to a webpage. Users can upload custom images from local directory to the webpage and view matches along with their respective match scores as a list.

**Video:** add link

**Tools:**

* **Server-side:** Python’s Flask API, Python OS library, IBM Watson visual recognition API
* **Client-side:** JavaScript, jQuery, AJAX, HTML5, CSS3, Bootstrap

**Techniques**

Figure : Process

* The webpage loads with Python Flask API serving an HTML file over REST API
* An HTML form takes in the file path of the local image that the user chooses
* A validation layer confirms appropriate file extension
* A synchronous AJAX call with image file payload is served to flask upon clicking on “Classify”
* The server endpoint receives this file and stores it locally temporarily
* The IBM Watson visual recognition API reads the image file and calls the classify function
* The temporary image file is deleted
* The response consisting of key value pairs of classes and scores is returned
* The webpage displays this response in a tabular form

**Data Sources**

* The images are sourced from google image search
* Irrelevant images like graphical designs, sketches, etc. are filtered out
* Around 200 images are selected for each class with varying background, multiplicity, cuts, levels of ripening, exposure, etc
* Around 100 images are selected of fruits other than the mentioned ones for negative class
* Around 10 images from each class are taken out as test sets

**Special processing requirements**

As mentioned

**Problems and solutions**

As mentioned

**Redoing experiments with more/different images**

As mentioned

**Summary of results**

As mentioned

**Assessment of the resulting system’s effectiveness**

As mentioned

**Improvements**

As mentioned

**References**

As mentioned