# Project Proposal

## Jordan Bell

## Introduction

I plan to research the possibility of generating arbitrary but consistent Unique IDs from images. I will implement this algorithm to the best of my ability, and, if feasibility and time permit, explore various applications of it.

## Base Aims

I will aim to create a function that takes a matrix of pixel data and outputs a unique value based on the traits of that data. The aim is so that, for example, the same value will be generated from different pictures of a company logo taken in different lighting conditions (different location, time of day, etc.)

The algorithm will aim to prioritise false positive results over false negatives. For example, a logo which looks similar may generate the same value, while an image of something completely different should not. I would consider these results acceptable.

## Extension Goals

The meat of this project is in the extension goals, which will explore applications of this algorithm.

I would hope to build a simple program for Android phones that will take photos, prompt appropriate cropping to focus on the subject, and perform the algorithm. Once a value is obtained, it will retrieve a URL from a server corresponding to that image-key. If this functions within reasonable accuracy, I believe it could be a pretty fun platform for encoding data in real-world objects, similar to QR codes. This could be an exciting prospect for marketing and communication.

## Technologies

The images will be obtained using my personal smartphone. I plan to write the algorithm in C/C++ using Cuda multi-threading image processing if necessary, for large numbers of samples. The extension will also be written in C/C++ and Java for Android applications.

## Tackling Scope

There are several factors that can make this goal more feasible. I will research dimension reduction techniques, apply methods to reduce the number of input pixel permutations, impose reasonable limits on input data, and require user guidance where possible.

The aim of this project is **not** to classify or identify objects. By generating arbitrary and anonymous values, we avoid the problems presented with identifying objects as part of an existing concept (cars, animals, people, etc.). For example, this algorithm may produce different values for images of a dog from different angles. Unlike image identification, the algorithm focuses solely on identifying 2-dimensional images of the same subject, not necessarily acknowledging or understanding what that subject is.

## Permutation Reduction

The most immediate issue I will come across is the sheer number of image permutations.

If the number of pixels in a resolution is n, and the number of colours possible for each pixel is c, then the number of permutations for that image is nc. Thus, the number of colours is the key factor in the permutation issue, with the resolution being a secondary contributor

It is proposed that we reduce the colors to 32 to best reduce the number of permutations, while still maintaining reasonable detail.

An important detail to note is that this is largely affected by colours surrounding the focus of the image. Therefore it is proposed that the user is given the option to crop the image to give focus to the subject. This will give a better range of colours for the subject itself.

For sample image input, I will use back-facing camera of a Samsung Galaxy S6 smartphone. To simplify the algorithm, I will limit the input image size to no more than 64 by 64 pixels. This will require automatic scaling down from the native 2988 x 5312 resolution of the camera.

Steps will have to be taken to ensure that these reduction steps are suitable and do not produce false negatives as a result.

## Example

The following are two images after reducing the resolution and number of colours. The algorithm should produce identical values for the two images. This will be built on intuitive criteria, which will be the main focus of my project. The “arbitrary but consistent” value will essentially be a serialised collection of traits that have, in themselves, been dimensionally reduced. The hope is that the intersection of all classes will hone in on a value accurately representing the key features that make these images “practically identical”.

 

## Timeline

### Week 1

**Background reading** on image classification techniques, existing research into technical definitions of image similarity, etc. Will build up a collection of sample images for testing the algorithm on.

### Weeks 2 – 5

Development of **composite criteria** contributing to an image’s unique ID. This will include testing and verification for each, using the test cases gathered in Week 1. Hopefully, as each criteria is developed, the number of matches will reduce false positives and avoid false negatives altogether.

### Week 6 - 7

If by this point an algorithm is working that can be used by an **application**, I will begin work on the Android phone implementation. I will set up the project to the extent that it runs on my phone, and allows me to take photos.

### Week 8 – 9

A **server** will be created that receives image data from the client Android app, perform the algorithm, lookup a corresponding URL (if one exists) and send a response to the client. I will improve the client app to notify the user of the server’s response, handling success and error states accordingly.

### Week 10

This week will be used for demonstration preparation and finishing of the remaining tasks.