

Evolving Optimal Histogram Parameters for Object Recognition

John Rieffel, Christopher DiLeo, and Bruce Maxwell^a

^aSwarthmore College, 500 College Ave, Swarthmore, PA 19081

ABSTRACT

Three dimensional color histograms are introduced as an effective means of object recognition. No globally optimal set of color histogram parameters is known, and the choice of data-set specific parameters is far from obvious due to the size of the search space involved. Evolution Strategies (ES), a form of Evolutionary Computation, are introduced as a method of optimizing histogram parameters specific to a known data set. An ES is implemented on a 22-object, 110 image database, and a 93% recognition rate achieved, a significant improvement over the 86% recognition rate of standard histogram axes. The results demonstrate the efficacy of ES and underscore the importance of the assumptions that histogram-based recognition methods are built upon.

Keywords: Object Recognition, Histogram, Evolutionary Computation, Evolution Strategies

1. INTRODUCTION

The task of object recognition is a difficult and long-studied problem in the field of machine vision, and the methods involved as varied as the applications. Almost all forms of object recognition require the machine in question to have or build information on defining characteristics of the objects in question. Feature detection methods, for instance, require an internal three dimensional model of an object's feature coordinates in order to perform an affine transform to retrieve pose information. Color distributions, because of their relative independence to variations in viewing angle and scale are another such defining characteristic. A particular method of representing color distributions for object recognition, the *color histogram*, is the approach that we use in this paper.

The efficacy of color histogram intersection methods has already been demonstrated in a variety of contexts (^{1 2}). Swain and Ballard, for instance, have used histogram intersection to effectively recognize objects from a 66-object database.¹ The strength of histograms is twofold: an object's histogram changes only slightly under varying poses, scales and occlusions, and histograms for different objects are often very distinct. As Swain and Ballard note, and Healy and Slater attempt to resolve, a primary weakness of histograms lies in their sensitivity to illumination.

While the color indices proposed by Swain and Ballard, proved highly accurate for their data set, there is no guarantee that these axes are by any means optimal for all data sets. Lacking a purely analytical means of determining optimal histogram parameters specific to a particular data set, finding them becomes an optimization problem. In this paper we introduce a form of Evolutionary Computation called Evolution Strategies (ES) as an effective means of optimizing color histogram parameters.

2. HISTOGRAMS FOR OBJECT RECOGNITION

Three dimensional color histograms are a compact way of representing a digital image's color distribution. A color histogram breaks a color's intensity into a discrete set of ranges called *bins* and counts the frequency at which each intensity range occurs within the image. Figure (2) shows red, green, and blue 20-bin histograms of the segmented image in Figure (1). A three-dimensional color histogram combines the axes of three separate histograms and counts their joint occurrence. A 3-D histogram is best represented as a three dimensional array, the indices of which are the bins of each color space.

Because the histogram of an object changes very little under varying poses, scales and occlusions (relative to the difference between histograms for different objects), an image of a particular object can be matched to an object

Further author information: (Send correspondence to Bruce Maxwell)

Christopher DiLeo: E-mail: cdileo@scs.swarthmore.edu WWW: <http://www.scs.swarthmore.edu/~cdileo>

Bruce Maxwell: E-mail: maxwell@swarthmore.edu WWW: <http://www.palantir.swarthmore.edu>

John Rieffel: E-mail: jrieffel@scs.swarthmore.edu WWW: <http://www.scs.swarthmore.edu/~jrieffel>



Figure 1. Original (left) and Segment (right) image

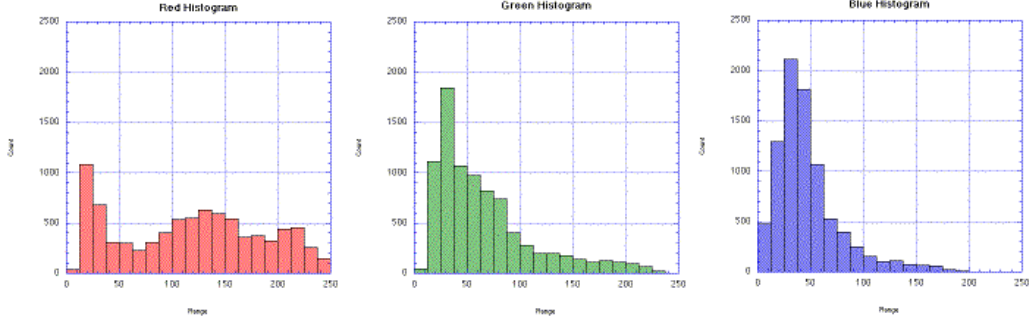


Figure 2. 20-bin Red, Green, and Blue Histograms of Figure (1)

within a database of object images by comparing their color histograms. The exact process involved in matching histograms is called *Histogram Intersection*.¹ The intersection of a model histogram, M with an image histogram, I is defined as

$$\sum_{j=1}^n \min(I_j, M_j), \quad (1)$$

which results in the number of pixels in the model that have corresponding pixels of the same color in the image. This value can be normalized by dividing by the number of pixels in the model histogram:

$$H(I, M) = \frac{\sum_{j=1}^n \min(I_j, M_j)}{\sum_{j=1}^n M_j}. \quad (2)$$

For a set of objects that we are trying to recognize, a set of object model histograms derived from images of each object. A match between a test image and an object can be found by intersecting the test image's color histogram with each of the model histograms. The image-histogram/model histogram intersection $H(I, M)$ with the highest value of corresponds to a best match for that particular image.

It is important to note three central assumptions about the images used in histogram recognition

Illumination Invariance: color distributions often change radically under variations in illumination, therefore it is important to ensure that all images in the database are under identical illuminations. While approaches toward reducing this dependance exist, most notably,² we will impose the illumination constraint for the scope of this experiment.

Pose Invariance: while most object's color distributions change very little under varying poses, some change considerably – we will largely confine ourselves to the former set of objects.

Object Size: the histogram intersection detailed in Equation (2) assumes that the model image and test image have the same number of pixels. If the test image has significantly more pixels than the model image, the test

histogram bin contents will consistently “outweigh” the model histogram bin contents – resulting in a false identification. Methods that segment objects of varying sizes from their background may produce such unbalanced histograms – an obstacle that must be addressed.

As we will see in the course of the experiments below, when these assumptions break down, histogram-based recognition suffers.

2.1. Histogram Axes

Any choice of colors or color combinations can be used to create three-dimensional objects. Merely using the red, green, and blue values of an image is rarely the best choice, because a change in white-light intensity will shift the entire histogram manifold along the three axes.

Previous work indicates a large dependence upon choice of histogram axes for object recognition from an image set. Swain and Ballard,¹ for instance, introduce opponent-axis parameters as follows:

$$\begin{aligned}rg &= r - g \\by &= 2 \cdot b - r - g \\wb &= r + g + b\end{aligned}$$

These axes were capable of correctly identifying objects from a database of 66 objects with 99% accuracy.

A generalized set of histogram axes can be represented as follows: we define the set of axis parameters P as a linear transformation from r, g, b space:

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} T_{1,1} & T_{1,2} & T_{1,3} \\ T_{2,1} & T_{2,2} & T_{2,3} \\ T_{3,1} & T_{3,2} & T_{3,3} \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (3)$$

The transformation matrix for Swain and Ballard’s opponent-axis, colors, for instance, would be:

$$\begin{bmatrix} rg \\ by \\ wb \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & -1 & 2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (4)$$

Each axis also has a integer number of bins associated with it.

The best choice of parameters for a given data set is far from obvious, and no purely analytical or mathematical methods of determining them are known. In this absence, the choice of histogram parameters becomes an optimization problem, but the range of possible parameters as defined by the range of the values for transformation matrix comprises an enormous search space. Adding the number of bins (discretizations) along each axis as a variable in the equation increases the scale of the search space even further. In order to find the best choice of parameters for a given data set, a technique capable of efficiently and effectively finding optimal solutions within a large search space is necessitated. We propose Evolutionary Computation as a solution to this optimization problem.

3. EVOLVING HISTOGRAM PARAMETERS

Abstractly, optimization of a particular problem involves searching through the space of potential solutions for a “best” solution to that problem. Problems with small search spaces can often be optimized using exhaustive search methods. Larger search spaces for which (differentiable) mathematical expressions exist can often be efficiently searched using techniques such as hill-climbing. But, for problems with large search spaces and for which only heuristic definitions of “goodness” exist, more novel search methods must be used. Evolutionary Computation is one such method.

Evolutionary Computation (EC) is a biologically inspired method of optimization and feature space exploration. The biological bases of the Evolutionary Computation are evolution and the Darwinian principle of natural selection, wherein only the most fit members of a population survive to reproduce. In this paper we will be focusing on the Genetic Algorithm (GA) and its cousin, Evolution Strategy (ES).

The fundamental unit of the GA is the genome - a string of values that describes a potential solution to the problem to be optimized. A genome can be evaluated via a heuristic *fitness function* that returns that particular solution's relative "goodness".

The basic Genetic Algorithm maintains an (initially random) population of genomes (also called individuals). The EC selects the most fit individuals of the population and then creates a new population of individuals from these "parents" by combining and modifying the parent genomes.

Creation of new individuals from parent genomes usually takes two forms: *mutation* and *crossover*. Mutation creates a new individual by probabilistically changing the values of a parent genome by a small and random amount. Crossover creates a new genome by combining two parent genomes. Conceptually, mutations result in new solutions near the parent solution, and crossover results in larger jumps through the solution space. The choice of mutation and crossover rates for a GA can vary depending on the type of problem to be solved.

A box chart of the evolution process is shown in Figure 3. These steps are repeated until a sufficiently fit individual is created, or a maximum number of generations is reached. This entire process process is analogous to an intelligent random search of the function's feature space: successive generations are variations upon the best solutions of the preceding generation and only survive if they offer an improvement. For a more rigorous mathematical exploration of Genetic Algorithms, including proofs of convergence, the reader is referred to ⁽³⁾ and ⁽⁴⁾.

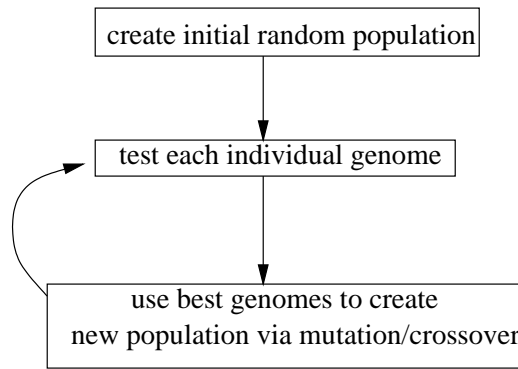


Figure 3. Diagram of Evolutionary Learning Process

3.1. Evolutionary Strategies

The standard Genetic Algorithm described above has several shortcomings when applied to certain problems, including histogram axis optimization. Most GA's are designed to use bit-strings as genomes, but the values of our histogram parameters are real-valued. Furthermore, GAs often perform weaker on variable spaces that exhibit *epistatis* - that is, when variables assumed to be independent are in fact correlated. This un-accounted-for correlation can drastically increase the convergence time of GA methods.⁵

A specialized form of GA, the Evolutionary Strategy (ES), is designed for real-valued variables and is more robust in the face of epistatis. Their strength lies in giving each variable its own standard deviation, σ , and evolving each σ along with the variable.

$$genome : \{(x_1; \sigma_1), (x_2; \sigma_2), \dots, (x_n; \sigma_n)\} \tag{5}$$

Mutation of each variable, x_i now takes the specific form:

$$x'_i = x_i + \sigma_i \cdot N(0,1) \tag{6}$$

$$\sigma'_i = \sigma_i \cdot e^{(\tau \cdot N(0,1))} \tag{7}$$

where $N(0,1)$ represents a randomly sampled normal number with mean 0 and standard deviation of 1, and τ is a learning rate parameter roughly proportional to $\frac{1}{\sqrt{2n}}$.

The standard $(\mu + \lambda)$ ES maintains a population of size $(\mu + \lambda)$. During each generation it selects the μ best individuals of the population and replaces the λ remaining individuals with offspring of the chosen parents.

As ⁵ demonstrates, ESs arrive at a solution much faster than standard GAs in situations that involve real-valued variables and suffer from epistasis.

For a more thorough description of Evolution Strategies we refer the reader to ⁶.

3.2. Defining Histogram Goodness

Quintessential to our process is a clearly defined fitness function, F , by which various histogram methods can be judged and compared.

In essence, a particular set of histogram parameters can be judged by the percentage of correct test-image to model-image matches it makes. The specific process is as follows:

For a set of images of the model objects, M , a set of model histograms $H(M_i)$ can be generated. Then, for a test image T that contains one of our objects, a corresponding test histogram, $H(T)$ can be created. The test histogram is intersected with each of the model histograms. If a given object's test image histogram best matches its corresponding model histogram, the match is considered correct. If this process is reiterated over an entire set of test images, goodness can be considered as the percentage of object test images that were correctly identified.

Specifically,

$$F = \frac{\text{number of correct identifications}}{\text{total number of test images}} \quad (8)$$

3.3. Evolution of Histogram Parameters

In the context of evolving best-fit histogram methods, an 'individual' is characterized by a unique genome consisting of a linear transformation matrix T of the vector $[r \ g \ b]^T$ to the vector $[P_1 \ P_2 \ P_3]^T$, where P_i is the i th histogram parameter as described in Section 2.1. Each genome therefore represents a unique *type* of histogram.

An initial population of N individuals can then be generated by assigning random values to each variable of the transformation matrix and each axis' binsize. Each individual can then be tested and scored using the fitness function F described in the previous section.

4. EXPERIMENTS

The image set used for these experiments was composed of 6 images each of 22 unique and chromatically complex objects (See Fig. 4). One image of each object was used as the model image, and the remaining 5, varying largely in pose from the model image, were used as test images.

All photos were taken with an Olympus digital camera under fluorescent lighting with a matte black backdrop and reduced to 160x128 resolution. Pixel intensity was thresholded to segment each object's image from the black background. Computations were performed on a dual 450 MHz Pentium II with 256Mb of RAM running RedHat Linux 5.2. Unless explicitly mentioned otherwise, all Evolution Strategies run were $(\mu = 6, \lambda = 30)$ with an initial σ of 2.5, an initial τ of 0.1, and a 10% chance of one-point crossover. With these parameters, each 30-member generation took approximately 10 minutes to run. A run of 100 generations took approximately 16 hours.

Figure (5) shows the progress of the average ES run. The scores of the minimum, maximum, and average individual of every generation is shown. The solid lines represent the scores of Swain and Ballard and standard RGB histogram parameters.

Swain and Ballard histograms achieve a recognition rate of 85.4% whereas RGB manages to score 86.3%. The recognition rate of the best evolved histogram after 100 generations was 93.6%, which corresponds to only 7 missed out of 110 test images. Appendix (A) contains the confusion matrix results of the histogram evaluation.

The missed objects are as follows:

Rainbow Mug: Three test images misidentified as hackysack.

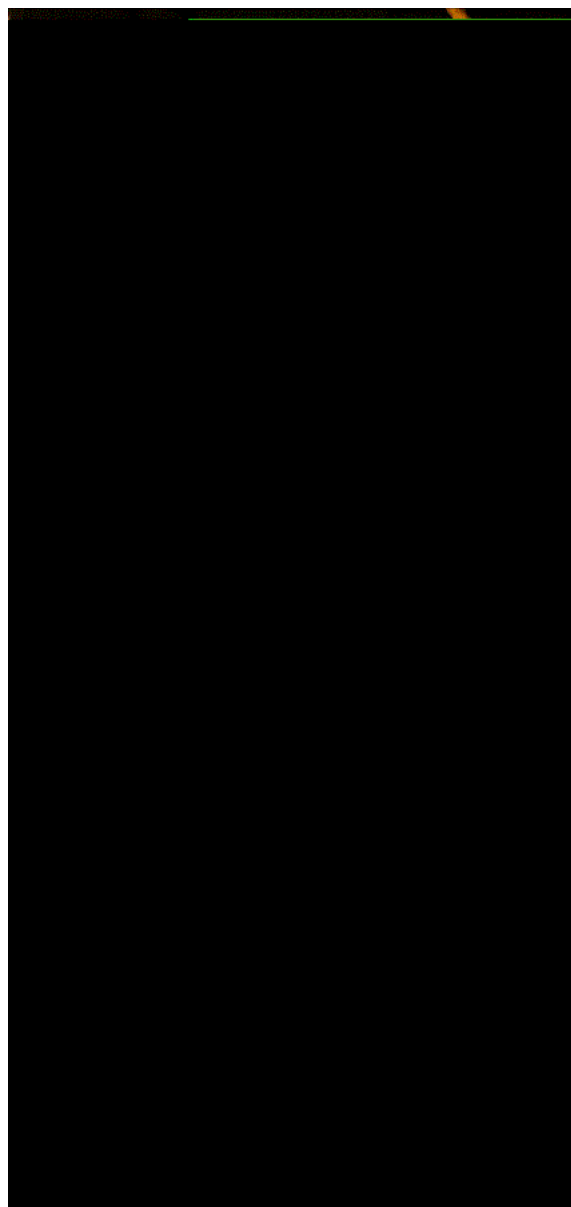


Figure 4. 22-Object Image Data Set

Plastic Cup: One test image misidentified as a core sample.

Author 1: One test image misidentified as “latte” cup.

Plant Food: Two test images misidentified as hackeysack.

Upon inspection, each of the misidentifications is a result of a breakdown of one of the assumptions made for histogram recognition.

Rainbow Mug: As Figure (6) shows, all three misidentified mugs were under a significantly different illumination than the model image. The difference in illumination is most likely the result of a shadow cast by the photographer. The basic white background of the mug is a dirty yellow in the three test images, and the yellow stripes in the model image appear orange. Under such a drastic change of color distribution, a misidentification is understandable. The hackeysack is the smallest of the 22 objects and, as a result, its histogram is most likely to be “outweighed” by the test image’s histogram as described in Section 2.

Plastic Cup: Again, the misidentified test image (Figure (7)) is under a different illumination than the model image and appears significantly greyer - enough so that its color distribution more closely resembles the core sample. In this case the color change was slight enough that the second-best guess was the correct answer.

Author 1: In this particular test image of the author (Figure (8)), he is turned sideways and, as a result, his blue t-shirt is occluded. As a result, his color distribution more closely resembles that of the latte cup. Again, the correct answer was the second-best guess.

Plant Food: The most difficult misidentification to justify, this may be a result of the change in the image’s color distribution under varying poses. As in the rainbow mug misidentification, the test image histogram “outweighs” the hackeysack’s histogram for a false identification.

5. CONCLUSIONS

As the experiments described above demonstrate, Evolution Strategies are effective at optimizing color histogram parameters for a specific data set. The recognition rate of the best solution after 100 generations was a significant improvement over the rates of both Swain and Ballard and RGB axis parameters. Furthermore, the majority of misidentifications were a result of test images that did not fit the assumed constraints of the recognition method.

The process could be refined in several ways. Most importantly, the method of histogram intersection used does not take into account histograms with a radically different number of pixels (as a result of object size and background segmentation). Normalizing each histogram with respect to the number of pixels in its respective image *before* intersection would improve the accuracy of the intersection method significantly. Furthermore, modifying the fitness function to reward solutions based upon the *rank* of the correct answer might improve the speed of evolution. Finally, the parameters of the ES evolution, especially the learning parameter τ and initial standard deviation can be fine-tuned to improve the overall performance of the ES method.

REFERENCES

1. M. Swain and D. Ballard, “Color indexing,” *Int’l Journal of Computer Vision* **11**, 7 1991.
2. G. Healey and D. Slater, “Using illumination invariant color histogram descriptors for recognition,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* , 1994.
3. D. Dasgupta and Z. Machalewicz, *Evolutionary Algorithms in Engineering Applications*, Springer, New York, NY, 1997.
4. M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, MA, 1998.
5. R. Salomon, “Increasing adaptivity through evolution strategies,” *International Conference on Simulation of Adaptive Behavior* .
6. T. Back, U. Hammel, and H.-P. Schwefel, “Evolutionary computation: Comments on the history and current state,” *IEEE Transactions on Evolutionary Computation* **1**, April 1997.

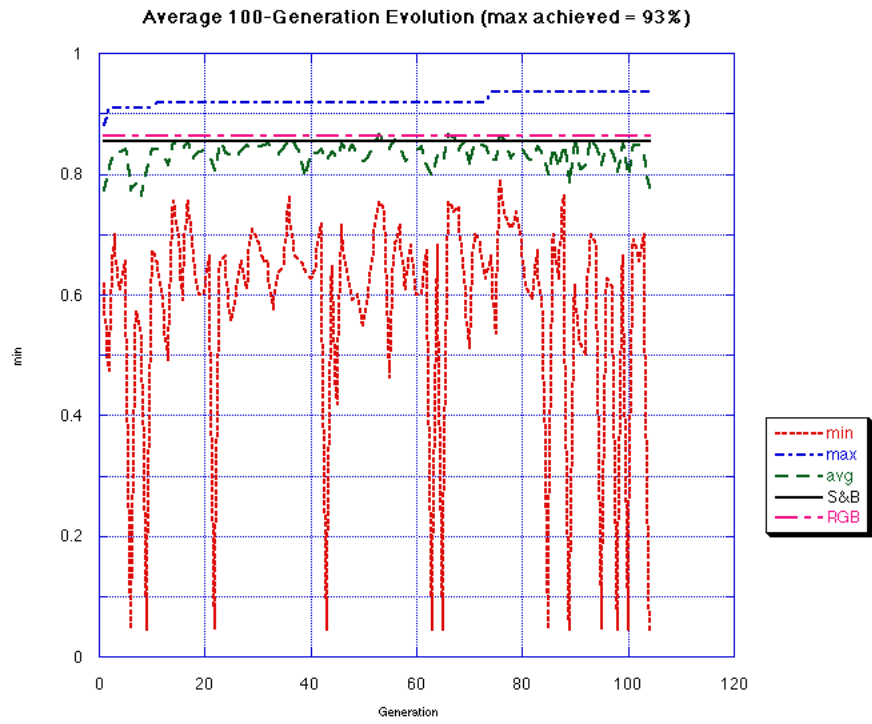


Figure 5. Progress of Average 100 Generation Evolution

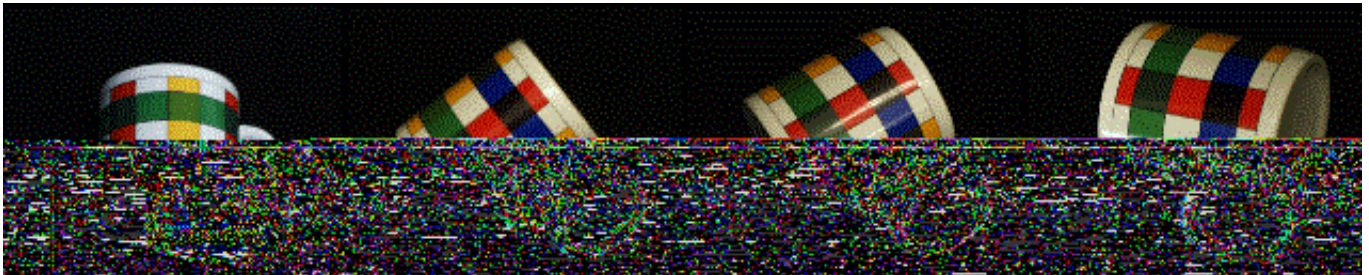


Figure 6. Model Image of Mug (left), and 3 Missed Test Images

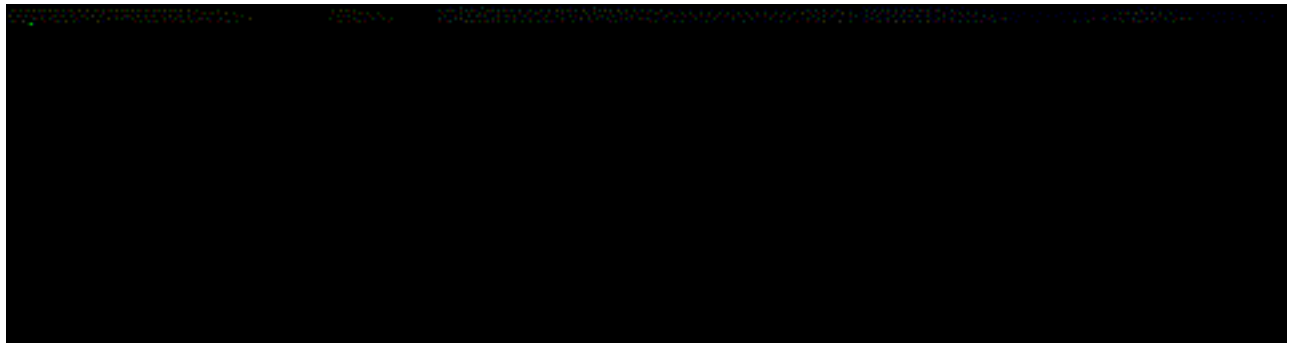


Figure 7. Model Image of Cup (left), Missed Test Image (center), and False Identification (right)



Figure 8. Model Image of Author (left), Missed Test Image (center) and False Identification(right)