Artificial Neural Network For Automated Prediction of Popularity of Digitized Images

David Oranchak
Roanoke, VA 24018, USA
doranchak@cox.net
Student, Neural Network Design, Dr. Milos Manic
http://husky.if.uidaho.edu/ee578s06/

Abstract -- The use of an artificial neural network to predict subjective human reactions to photographs is explored in this paper. In particular, the neural network is used to determine the factors that distinguish subjectively good photographs from subjectively bad photographs. Data used for training come from the photograph sharing web site Flickr.com, which currently contains over 100,000,000 photographs and a estimated social network of approximately 1,000,000 registered users. The site systematically determines which photographs are the most popular and have the most appeal to the most users.

I. INTRODUCTION

A human being uses subjective reactions, such as personal taste and artistic appreciation, to judge the overall quality of an image. These reactions are unpredictable and depend largely on the uniqueness of the individual human being observing the image. However, some images trend towards more universal acceptance among varying subjective tastes. These images are more likely to be considered interesting, artful, or high quality by a larger group of people. The social-network-driven web site Flickr.com has amassed over 100,000,000 photographs from a user base of approximately 1,000,000 users. The company that created the site has established a ranking system that automatically determines the "interestingness" of user-submitted images in comparison with other user-submitted images. For an example of this, see http://flickr.com/explore/interesting/2005/01/. These rankings are generally based on objective criteria, such as the frequency of user visits to specific images. Thus, the determination has already been made of which images have reached a more universal acceptance among the varied tastes of the user base. A computer algorithm that can use this information to predict the "interestingness" of arbitrary images has high potential in many areas. For example, advertising or marketing agencies can run stock graphics through such an algorithm to determine which images are more likely to achieve higher public acceptance. A consumer digital image management product can use such an algorithm to mine the user's substantial digital image collection to filter out the most interesting images. Image search engines can rank image results with such an algorithm so users are viewing the most interesting results. Imaging algorithms or "computer art generators" can create galleries of artificially-generated images that have a high probability of having characteristics that are interesting to people.

The mechanism to obtain the training data from Flickr.com is described in this paper. Flickr.com provides a web services interfaces that enables software developers to access the data from the Flickr.com site. This paper explores image data analysis techniques to determine which data are useful as inputs for an artificial neural network to make determinations on subjective image qualities. This paper also discusses the selection criteria for the type of neural network used, and how its operational parameters are determined.

II. SURVEY

Neural networks research in the image analysis field is extensive and covers many different areas of interest. The specific problem this paper attempts to solve appears to be unique. However there are several solutions of interest that share some characteristics that are useful to this paper.

Ekman and Friesen et al utilize back-propagated feed-forward neural networks to train a system to classifying facial actions and expressions [1]. In their solution, several approaches individually produce useful results, and the combination of approaches yields a success rate of 92%. Of interest therein is the "holistic spatial analysis" which uses gray level images as training data in a three-level neural network with ten hidden units and six output units. The network uses a soft transfer function, and the output is determined using a winner-take-all scheme. Training is accomplished using conjugate gradient descent. The other techniques used in the paper are feature-specific to faces, and are not considered here, since the problem this author presents is not feature-specific. However, the technique of combining multiple approaches is significant to our ability to detect the interestingness of photographs, as our own judgment of the interestingness of an image is intuitively suspected to be based on more than one criteria.

Rowley, Baluja, and Kanade utilize a "retinally connected" neural network to examine portions of images to decide if they contain a human face [2]. Their system chooses among a variety of networks to improve the performance that may be limited by using a single network. Presented in their paper is a challenge that is relevant to us: for the purpose of training the neural network to detect faces, it is easy to get a representative sample of images which contain faces, but much harder to get a representative sample of images that do not contain faces. To address this, Rowley, Baluja, and Kanade use a technique of "bootstrapping" non-face images by selectively adding images to the training data set as the training is performed. Their use of bootstrapping combined with multiple neural networks resulted in improved accuracy. The topology of the neural network used reflects some general sense of the facial features being detected. For example, a group of hidden units in the network is arranged in stripes in order to detect mouth features in the input images. Similarly, a group of hidden units arranged in smaller squares is used to detect eye features.

Gallagher and Deacon present a neural network used to classify mineralogical samples using x-ray spectra [3]. Input data is presented to several varieties of neural networks in the form of histograms representing x-ray spectral data obtained from minerals analyzed by x-ray spectral machines. A 100% success rate was obtained via experiments using the Backprop and Quickprop neural network algorithms. Of interest to this author is the use of histogram spectra data to conduct the classification. One of the factors in classifying the interestingness of a photograph is the color distribution. Experiments herein will represent the color distributions via histogram data.

III. CLASSIFICATION OF POPULAR IMAGES

In July of 2004, the site Flickr.com

began ranking user-submitted photographs using a methodology of quantifying the measured user activity centered around individual images. Data used to determine the popularity of a photograph include: measurements of user clickthrough from web sites that link to the photograph, the quantity and creation time of user-supplied comments, the number of Flickr users that tag the photograph as "favorite", and the types of tags used to describe the photograph. All of these data are human-generated, and Flickr generates a quantitative ranking using an undisclosed algorithm. With this approach, the Flickr.com site is able to display a collection of the most interesting photographs as determined by user activity. Generally this results in automatically detecting interesting, high quality photographs. Figure 1 shows an example.

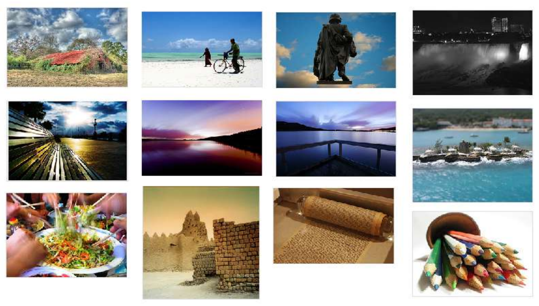

Figure 1 - Flickr.com's algorithm determined that the above were some of the most interesting photographs on February 1 2006 and February 2 2006.

Flickr clusters the rankings based on the dates the photographs were uploaded to the site. Thus, a photograph with the top popularity ranking for one day will have equal popularity ranking with a photograph with the top popularity ranking for a different day. Since the photographs are ranked using a quantitative approach, there is likely to be a way to compare relative rankings between different days, but this is not yet exposed on the Flickr site or API. The ranking is obtained from the Flickr site using the Flickr web services API. Flick exposes a remote method call specifically designed to return photographs ordered by the popularity ranking. Thus interestingness is quantified based on the position each photograph appears in the list. For the purposes of this author's experiments, photographs are limited to certain sample areas. A sampling is taken from the top of the ranked photographs. Then, a more sparse sampling is taken for less popular photographs in order to get a useful representation of varying levels of popularity. A minimum popularity is also selected. The number of top rankings to sample, degree of sparseness of intermediate ranking samples, and minimum popularity threshold are variables used in a trial and error approach during experimentation. The minimum popularity threshold is used to scale the output of the neural network to a unipolar or bipolar value. For example, in the unipolar case, the value of 0 represents lowest popularity, and 1 represents highest popularity. Additionally, categories of popularity are defined to encourage clustering during the training. For example, category 1 is "Very Interesting"; category 2 is "Somewhat Interesting" and category 3 is "Not Interesting." One observation is that the user-based measurement of a photograph's popularity may not detect high quality photographs that are currently marked with much lower popularity on Flickr. This author hopes that the general trend (higher rankings imply higher quality and lower rankings imply lower quality) will outweigh these occurrences on average.

Several forms of input are used to perform neural network operations for this classification problem. Each form is intended to isolate specific characteristics of the image data. This produces a higher variety of training data that are combined to generalize the classification results. One input form is the original raw photograph data. This data is reduced to a 75x75 pixel size to improve the performance of the neural network simulation. Experiments using higher-resolution imagery may be performed but will require smaller sample sets. This training data attempts to approximate a popularity mapping function based solely on the pixel data that comprises the raw images. Experimentation does not rely solely on this technique due to the possibility of *overfitting* the neural network during the training. For example, given a set of 10 popular images and 10 unpopular images in the training data, the neural network may accurate in classifying the images, but only based on detecting those specific images and not any other factors leading to their popularity. This observation is also dependent on the sample set used.

Another form of input is the histogram representation of color distribution in the image data. This form breaks down into three categories. The first is a simple RGB (red/green/blue) count of the image data. For example, a photograph may be comprised of 20% red, 30% green, and 50% blue. Another category is the one-dimension color distribution histogram. Each color channel is separately analyzed into a histogram showing intensity versus channel value. Three-dimensional color distribution histogram is another category of histogram representation. The three-dimensional color histogram shows color intensity as a function of all three of the RGB channels. However, the three-dimensional histogram input has a space cost of 16,581,375 elements for the full histogram representation unless a sparse 3x3 matrix is used. The goal behind the color histogram techniques is to determine if the neural network can predict popularity as a function of color values. If successful the algorithm will determine if photographs that contain specific color distributions are more likely to be popular than other photographs.

An important consideration in the analysis of photograph quality is the luminosity or lighting values of the photograph. The luminosity can be approximated by conversion of the original photograph into a gray scale representation. The RGB color channels will thus be equalized (monochromatic) and a histogram analysis will express luminosity , or light intensities, rather than color distribution. Using these data in training will determine if it is possible for a neural network to predict popularity based on light intensities. Specific distributions of lighting values may increase the probability of high photograph popularity.

Another input form is the use of texture analysis via gray level co-occurrence matrices. Co-occurrence matrices are used to compute several factors used to quantify texture properties [4, 5]. These properties include: contrast (or intertia), energy, entropy, symmetricalness, correlation, dissimilary, and homogeneity (or inverse difference moment) [6]. Quantifications of these properties are used to express spatial relationships, tone and structure of an image. The texture-based approach to the training set includes both the examination of input data resulting from texture analysis, as well as the inclusion of the entire co-occurrence matrix on scaled images or image subsets. Experimentation will explore the effect of using solely the matrix-derived properties above versus allowing the neural network to derive new properties based on the co-occurence matrix and the desired popularity rankings.

IV. EXPERIMENTAL METHODOLOGY

Obtaining Photographs

The first step was to obtain the raw image data set from Flickr used for training. Flickr makes available a web-based API for directly accessing their exposed services. To simplify the activity of accessing this API, experiments herein used **flickrj**, a free Java-based wrapper around the Flickr API. Flickr organizes popular photographs by date. Three experimental sets of photographs were obtained and are described below.

Four main experiments are conducted. The first experiment performs neural network training against a sample data set of 559 images obtained from Flickr.com. These images are categorized into three distinct categories. The first category, "Very Interesting", consists of photographs whose popularity ranking falls within to top 25. Nine different sample days are used. The sample days are: August 9, 2004 through August 12, 2004 inclusive, and August 15, 2005 through August 19, 2005 inclusive. This category contains of 221 distinct photographs. Figure 2 depicts the top 25 photographs from one of the nine sample days.

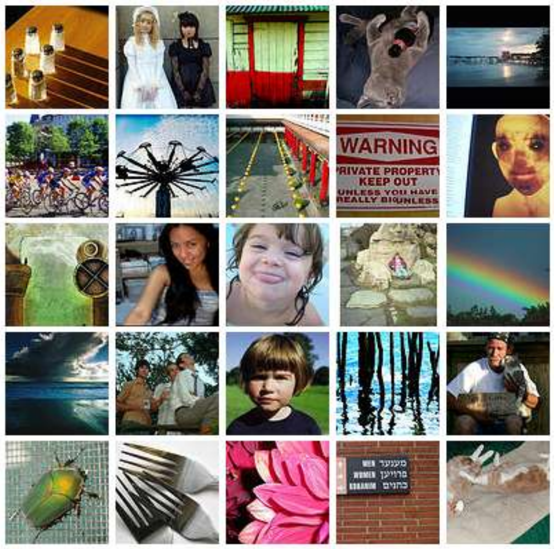

Figure 2: Top 25 Flickr.com photographs from August 11, 2004. These photographs are in the "Very Interesting" category.

The second category, "Somewhat Interesting", consists of photographs whose popularity ranking falls within the top 300 and 500, inclusive. These samples were taken from the same nine days. This category consists of 88 photographs. Figure 3 depicts 11 photographs in this category from one of the nine sample days.

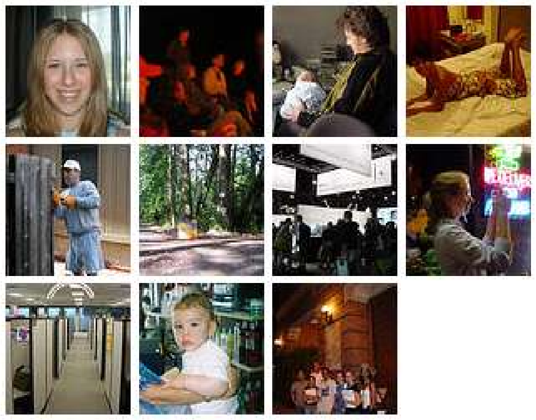

Figure 3: Eleven sample photographs from the "Somewhat Interesting" category.

The final category, "Not Interesting", consists of photographs that do not fall in the popularity ranking system on Flickr.com. At the time these experiments were conducted, Flick.com limited its popularity rankings to 500 selected photographs per day. Once the 500 photographs are ranked for a given day, the remaining photographs are considered unranked, and thus "not interesting". This category contains 250 images for the network training. Figure 4 depicts a sample of photographs from this category.

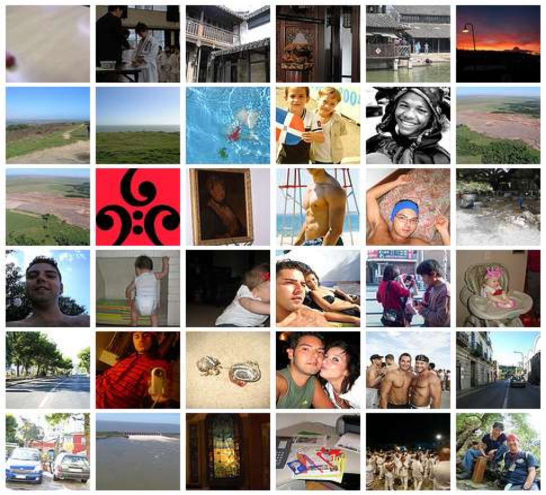
Figure 4: Thirty-six sample photographs from the "Not Interesting" category.

The second experiment uses the neural networks trained in the first experiment to attempt to predict popularity clusters for another image data set with known popularity rankings. The data set contains 2381 images sampled from 67 different days. The sample days are: July 1, 2004 through July 31, 2004 inclusive, August 1, 2004 through August 12, 2004 inclusive, August 5, 2005 through August 28, 2005 inclusive, and April 2, 2006. The popularity categories are known in advance. The "Very Interesting" category contains 1373 samples. The "Somewhat Interesting" category contains 557 samples. The "Not Interesting" category contains 451 samples. Figures 5, 6, and 7 depict samples from these categories.

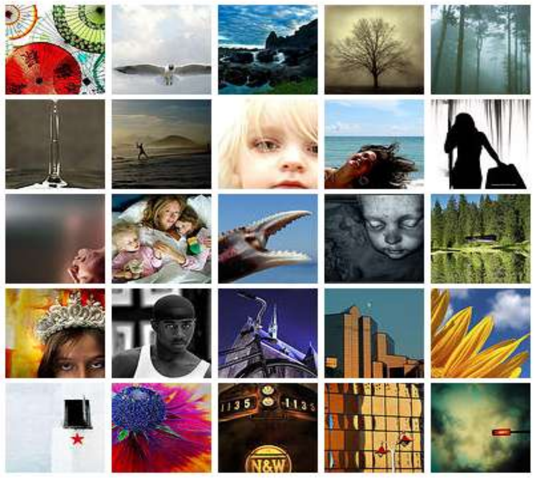
Figure 5: Twenty-five "Very Interesting" photographs from August 15, 2005.

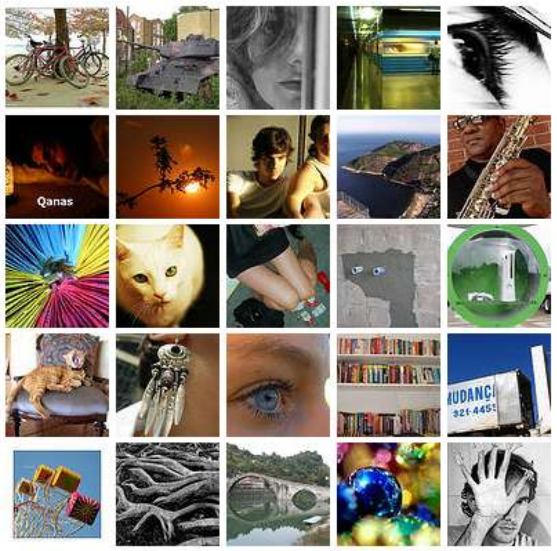
Figure 6: Twenty-five "Somewhat Interesting" photographs from August 15 through 17, 2005.

Figure 7: Twenty-five "Uninteresting" photographs in the test set for the second experiment.

The third experiment runs a sample set of 250 photographs through the neural networks trained in the first experiment. These photographs have an unknown popularity ranking. Additionally, they were all very recently uploaded by users to the Flickr web site at the time of the experiment. Figure 8 depicts a sampling of these photographs.

Figure 8: Twenty-five photographs from the 250 images used in the third experiment.

The fourth experiment uses the trained neural networks to attempt to locate interesting photographs from this author's collection of photographs. A sample set size of 2912 was used.

Processing Photographs Into Input Data

For each experiment, every photo is processed to produce a set of image metadata used as input data for the neural networks. The neural network topologies are described in a separate section below. The types of input data used are as follows:

- Raw pixel data, RGB, 10-by-10 pixels
- Raw pixel data, Gray scale, 20-by-20 pixels
- One-dimensional color histogram, RGB
- One-dimensional color histogram, Gray scale
- Color percentages, RGB
- Color percentages, Gray scale
- Texture metadata

Practical limitations of neural network topologies reduce the amount of information that can feasibly by used as input data. Thus, in the case of raw pixel data, the original images must be down sampled into smaller images. Flickr.com provides multiple downloads for each photograph. The medium and small downloads are used for this paper's experiments. The medium image size has a maximum width and height of 500 pixels. The small image size has a width of 75 pixels and height of 75 pixels. A side effect of these formats is that the small image size is not truly representative of the medium image size, because the small image size has an aspect ratio of 1:1, whereas the medium image size can have any aspect ratio, with the constraint that neither the image height nor width may exceed 500 pixels. Thus, the use of the small image size as input often results in a very loose approximation of the original image. Further, for network training, the full 75-by-75 raw pixel data produces a very large input vector. For gray scale image data, the input vector has a magnitude of 5,625. For RGB image data, we require three output channels each with a magnitude of 5,625, thus the input vector has a total magnitude of 16,875. For these reasons the input data is further reduced to image sizes of 10x10 for RGB images, and 20x20 for gray scale images. Input vectors for RGB and gray scale then have a magnitude of 300 and 400, respectively.

```
37 22 76 95 11 25 90 -126 -119 -101 30 10 -119 -109 22 4 88 107 -108 -87 22 4 96
-125 26 29 107 104 -113 -89 22 11 67 -58 106 -77 -90 86 -106 -82 -93 106 119 -25
111 -86 -83 93 -93 -75 -49 -51 -57 -62 107 47 104 -103 -81 -69 -56 -43 -93 -124 -116
79 126 -73 -70 -62 -81 -54 -67 -90 67 69 -98 -83 -82 -75 -90 -84 -103 107 81 54 115
-88 -82 -92 -104 -91 -99 122 102 72 124 -87 -90 -99 46 22 66 83 1 12 66 95 110 -128
39 9 -128 -118 19 0 75 84 125 -110 30 5 89 126 30 35 108 99 121 -111 30 10 59 -62
113 -64 -78 94 -126 -102 -85 105 108 -36 111 -77 -72 101 -114 -96 -41 -57 -75 -82
97 43 103 -107 -104 -92 -48 -51 -116 103 115 62 107 -94 -95 -87 -73 -64 -96 -127 33
41 -127 -116 -111 -104 -82 -80 -106 96 64 34 93 -111 -106 -116 -96 -87 -102 111 85
52 102 -110 -114 -123 45 14 54 71 0 0 38 57 67 85 38 4 119 -127 12 0 56 53 83 104
32 0 83 123 33 35 100 80 84 108 32 8 56 -61 121 -56 -80 81 97 121 -83 103 106 -32
119 -72 -78 86 111 -127 -38 -57
-77 -81 98 40 85 120 121 -123 -45 -53 -122 97 110 46 75 117 -126 -118 -70 -65 -102
120 21 19 87 86 115 122 -71 -77 -115 76 34 0 56 111 116 106 -85 -84 -111 91 55 17
65 112 108 99
```

Figure 9: Sample input vector for 10x10 RGB raw pixel data.

```
50 32 32 32 6 39 92 56 4 3 8 2 49 100 87 83 96 115 122 124 45 26 33 24 44 115 127
82 8 2 3 2 51 106 102 101 108 127 -122 -122 38 25 33 15 77 -68 -88 99 7 1 1 5 51 97
93 84 118 -117 -111 -114 32 31 30 13 69 -66 -65 91 19 12 4 8 50 97 103 91 119 -113
-106 -112 23 35 21 17 33 -121 -59 84 23 21 6 9 47 97 117 98 116 -113 -104 -111 20
30 13 22 9 88 -59 110 39 64 67 73 92 118 112 59 114 -110 -98 -106 32 24 13 25 15
84 -62 -94 75 -126 -100 -90 -89 -94 124 34 113 -108 -92 -99 48 21 18 27 32 101 -65
-51 79 -111 -81 -72 -75 -80 -119 38 110 -108 -90 -97 -78 -124 76 39 64 -105 -41 -40
94 -94 -33 -42 -58 -40 -88 53 124 -101 -84 -89 -46 -59 -84 -125 114 -95 -35 -18 120
119 -124 -117 -122 -120 120 80 -113 -92 -85 -93 -74 -57 -36 -52 -89 -93 -82 -88 127
72 34 50 86 109 -128 -109 -97 -87 -89
-97 -61 -63 -50 -49 -69 -82 -101 123 -96 104 35 27 83 -127 -111 -102 -98 -90 -91 -97
-47 -60 -55 -51 -65 -80 -109 106 -107 127 74 56 104 -109 -109 -117 -106
-93 -90 -93 -68 -64 -48 -53 -85 -108 125 94 83 78 66 78 117 -115 -109 -101 -109 -95
-89 -91 -61 -59 -54 -73 -106 -113 -114 122 70 44 39 73 110 123 -122 -104
-109 -98 -94 -95 -75 -74 -65 -70 -84 -85 -101 116 90 37 17 56 102 127 -117 -109
-110 -103 -101 -100 -69 -96 -74 -70 -113 -112 -108 99 59 57 46 45 96 -119 -123
-110 -109 -105 -109 -113 -83 -88 -79 -84 -111 126 103 67 59 54 40 37 86 -128 126
-117 -111 -108 -114 -119 -106 -88 -89 -95 -95 -116 106 91 90 77 57 48 92 -123 -123
-112 -113 -112 -119 -126 -113 -94 -98 -102 -88 -96 -118 -120 -128 108 81 65 100
-116 -116 -108 -113 -113 -123 125
```

Figure 10: Sample input vector for 20x20 gray scale raw pixel data.

The raw input data is scale-dependent and position dependent, which limits the ability to analyze input images and compare them to other images. Thus we conduct color analysis. Histogram computations are performed on the color distributions of the input images. Again an input data size concern arises when using color histogram data. In the case of three-dimensional color histogram data in the RGB color model, every combination of red, green, and blue values must be included in color counts. Since there are 256 possible color intensities in each of the RGB channels, the total number of combinations of colors is 16,777,216. This is impractical for the neural network learning algorithms. To reduce the magnitude of the input vector for color analysis, color metadata is limited to one-dimensional histograms and color percentage counts. In the case of one-dimension histograms for full-color RGB analysis, the input vector consists of three channels, each containing 256 entries, for a total magnitude of 768 inputs. In the case of one-dimensional histograms for gray scale analysis, the input vector consists of only one channel. In the case of color percentage counts, the image color data is split into eight regions. Each region is designated by color name: black, blue, green, cyan, red, magenta, yellow, and white. A color counting procedure determines the percentages of the image color data for each color category. The color percentages provide a rough sketch of the scale-invariant color distributions in the input image.

```
3 0 2 2 3 3 2 2 2 2 4 9 2 3 4 6 4 5 6 9 10 11 6 15 6 4 9 11 16 6 10 14 11 9 21 13 10 20
14 11 12 3 16 12 11 11 9 15 18 13 15 10 12 23 14 27 19 28 16 27 31 23 24 29 29 36
29 34 42 38 39 36 40 47 58 51 65 79 46 73 74 73 82 96 77 114 130 126 134 149 154
148 180 202 185 222 242 258 270 305 378 456 513 672 892 1112 1498 1925 2429
3024 3578 3977 4457 4831 4867 5398 5661 5928 6321 6600 6802 7050 6790 6454
5585 4885 4381 4107 3875 3618 3375 2840 2284 1807 1519 1319 1252 1153 1286
1347 1359 1413 1349 1322 1240 1140 1051 991 957 951 1010 948 919 860 876 846
791 715 579 528 518 455 455 387 383 382 323 299 241 248 237 211 181 197 176
170 146 163 146 146 134 126 142 124 104 116 120 162 249 381 425 552 946 1517
1519 2041 1455 1078 1231 1264 876 813 926 1175 1238 934 437 391 298 118 111
35 17 15 13 14 7 10 11 12 6 12 11 6 4 6 6 5 2 3 6 4 2 2 6 2 2 7 2 5 2 2 3 2 4 0 1 2 1 2
2 1 0 2 1 3
```

Figure 11: Sample input vector for gray scale, one-dimensional histogram data.

```
0.30002847 0.24680583 7.324219E-4 0.18839519 0.033996582 1.4241536E-4
0.018941244 0.21095784
```

Figure 12: Sample input vector for color counts of an RGB image.

The last input vector used for neural network training is texture-related metadata. The follow quantitative data [5, 6] are computed for each input image:

- Contrast

- Correlation (also called Intertia)
- Dissimilarity
- Energy
- Entropy
- Homogeneity
- Correlation Matrix Sum
- Symmetry

For RGB images, the above values are computed for each of the three color channels, resulting in an input vector magnitude of 24.

```
90926626 0.9999999999999951 3970046 1112022826 4708818.221576486
281642.73953644 748250 1 83762742 0.9999999999999989 3827710 892331760
4661049.616140028 286428.5259828772 748250 1 83712860 0.9999999999999941
3901552 527959020 4309677.930229572 280554.7784524056 748250 1
```

Figure 13: Sample input vector for texture analysis of three-channel RGB image.

Each of the inputs required for network training is obtained by use of a free Java image processing library called **JIU**. JIU computes color and light intensity histograms, performs image transformations, and performs texture analysis. **ImageMagick**, a free image processing utility, is used to convert input images into gray scale images to be used to represent luminosity. Experiments are conducted on input files generated by JIU and ImageMagick.

Once accurate training results are established for the input vector types decribed above, they are tested against the unknown data set of photographs. These photographs are obtained from other dates showing popularity rankings on Flickr. These photographs are not in the original training data set. They are associated with a popularity category based on the quantified popularity ranking on Flickr, thus an error rate is directly computed from the experimentation. The trained networks are then applied to sets of photographs that do not have popularity measures. Thus, experimental results can only be elaborated using subjective measures.

The results of the accuracy obtained from the experiments above is analyzed to pick out which attempts can be used together as a combined approach to predict the popularity of unknown photographs. These experimental results give insight into a direction for further experimentation to improve the success rate of these techniques.

Neural Network Architecture

Initial experiments were conducted on feed-forward networks using several back-propagation algorithms. Further experimentation resulted in a stronger focus on the use of counter-propagation algorithms for training. These neural networks were simulated using **Joone (Java Object Oriented Neural Engine)** and **JavaNNS**, the Java-based successor to **SNNS (Stuttgart Neural Network Simulator).**

Joone is a neural network framework designed to aid researchers, professional users and enthusiasts in creating, training and testing a variety of neural networks [8]. It supports both supervised and unsupervised learning, as well as modular (hybrid) networks. The supervised learning algorithms it supports are feed-forward, recursive, time delay, standard back-prop, and resilient back-prop. The unsupervised learning algorithms it supports are Kohonen SOMs and Principal Component Analysis. The Joone package features modular components used to build and implement a large variety of network designs.

JavaNNS provides a robust toolset of neural network training algorithms for simulating a large number of types of neural network configurations [7]. Because it is based on SNNS, JavaNNS supports a higher number of training algorithms and network configurations than Joone. Initial experimentation was conducted using Joone, and final experimental results were created using JavaNNS.

V. EXPERIMENTAL RESULTS

Joone: Backpropagation Training

Initial experiments were conducted in Joone using feed forward networks employing backpropagation training algorithms. The input data described in the previous section was used to train these networks. Outputs were defined as a unipolar scaled representation of the quantified ranking. A value of 0 represents the lowest popularity rating. A value of 1 represents the highest popularity rating. Seven networks were created, corresponding to each of the image metadata categories. These experiments resulted in high error rates using the input data described in the previous section. This was due to the difficulty in determining the number of hidden layers, and number of neurons per hidden layer that would result in successfully trained networks. Also, the high dimensionality of the inputs and high number of input patterns contributed to the difficulty during training. This path of experimentation was abandoned in favor of pursing Kohonen unsupervised training and counter-propagation network (CPN) training in JavaNNS. CPN was quicker to train and more straightforward to implement.

Joone Kohonen Unsupervised Training

Exploratory experimentation was conducted in Joone using the Kohonen unsupervised training algorithm on the texture metadata of 559 input images with known popularity rankings. Twenty-four input nodes were used, and five winner-take-all output nodes were used. This number of output nodes was selected arbitrarily to discover any naturally-forming clusters as a result of the Kohonen training technique. As a result of the training, only three clusters formed based solely on the texture data. Figure 14 tabulates the clustering results on the 559 input patterns.

Cluster#	***# of images***	***# of Very Interesting***	*avg rank*
1	37	12	0.36
2	297	125	0.52
3	225	84	0.45

Figure 14: Clustering results on 599 input patterns.

This result indicates a weak inclination of cluster #2 to identify a majority of higher-ranked samples. However, the average ranking of the cluster (0.52) indicates this cluster includes many samples with lower rank. Further experimentation, particularly using different numbers of output nodes and different types of input metadata, may produce more useful results. Without further experimentation, it cannot be said that cluster 2 will again identify the majority of higher-ranked samples. A Gaussian output layer may also be of interest to identify trends in the input data. The remaining experiments herein, however, focus on CPN as it is more successful at creating a "memory" of input metadata provided by the images used for training.

JavaNNS Counterpropagation Network

Training was performed in JavaNNS using the counterpropagation training algorithm. The network architecture consists of an input layer containing ***i*** nodes, a hidden Kohonen layer containing ***k*** nodes, and an output layer containing two nodes. Figure 15 illustrates this architecture.

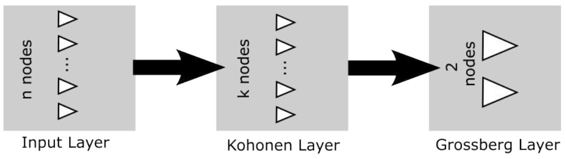

Figure 15: CPN architecture.

Five networks were successfully trained. Each network corresponds to one of five input vector types (raw RGB 10x10 pixel data, raw gray 20x20 pixel data, RGB 1D histogram, gray 1D histogram, and texture metadata). JavaNNS could not perform training on the color percentages input vectors. The reason is unclear; one possibility is that there is no good mapping between the 8-dimensional input vector and the two bit output. Another possibility is misconfiguration of the network and/or input pattern files.

The input layer size is dependent on the pattern width of the metadata used as input These are tabulated as follows:

- Raw RGB 10x10 pixel data: 300
- Raw gray scale 20x20 pixel data: 400
- RGB 1D histogram: 768
- Gray scale 1D histogram: 256
- Texture: 24

In each case, the hidden Kohonen layer is set to 559 nodes, to correspond to the number of images in the input set.

The output of the CPN consists of two nodes whose values represent one of three states: [0 0] for "Very Interesting", [1 0] for "Somewhat Interesting", and [1 1] for "Not Interesting". The choice of only three categories was done to reduce the computational overhead of larger neural networks, thus speeding the training. JavaNNS assigns random weights to the network during initialization, rather than directly computing the weights needed to activate the appropriate winner-take-all neurons in the Kohonen layer. As a result, the training performance is very dependent on the number of nodes in each layer.

CPN training of the five networks, each using 559 training patterns, resulted in sum-of-squares error below 0.01. The 2381 patterns belonging to the test data set with known popularity rankings were applied to each of the five trained networks. Accuracy percentages, calculated as number of correct outputs divided by number of patterns, are tabulated below:

- Texture: 51%
- Raw RGB: 47%
- RGB histogram: 53%
- Gray scale histogram: 51%
- Raw gray scale: 48%

The results indicate a poor ability of the networks to strictly identify specific ranking clusters. The breakdown below of accuracy percentages by cluster shows some improved results:

- Texture, Very Interesting: 50%
- Texture, Somewhat Interesting: 32%
- Texture, Not Interesting: 76%

- Raw RGB, Very Interesting: 46%
- Raw RGB, Somewhat Interesting: 24%
- Raw RGB, Not Interesting: 79%

- RGB histogram, Very Interesting: 46%
- RGB histogram, Somewhat Interesting: 32%
- RGB histogram, Not Interesting: 80%

- Gray scale histogram, Very Interesting: 54%
- Gray scale histogram, Somewhat Interesting: 26%
- Gray scale histogram, Not Interesting: 75%

- Raw gray scale, Very Interesting: 54%

- Raw gray scale, Somewhat Interesting: 24%
- Raw gray scale, Not Interesting: 81%

The results indicate a high accuracy in determining that a given image falls in the "Not Interesting" category. Further analysis indicates that in a majority of cases (70% and higher), the neural network will produce an output of "Very Interesting" accurately, and the combination of outputs to filter out false positives increases the accuracy significantly, at the expense of causing the neural network to overlook photographs known to be in the "Very Interesting" category. This analysis is tabulated by defining a threshold. This threshold indicates how many of the five networks have classified a given image as "Very Interesting." For example, a threshold of two indicates that at least two of the five networks have classified an image as "Very Interesting". As this threshold increases, accuracy increases while the total number of detections of known "Very Interesting" samples decreases. These results are tabulated as follows:

Threshold: 1
- Known "Very Interesting" photographs detected: 1241
- Percentage of total number of known "Very Interesting" photographs (1373): 90%
- Number of false positive detections: 577
- Total number of detections: 1818
- Error rate: 32%

Threshold: 2
- Known "Very Interesting" photographs detected: 988
- Percentage of total number of known "Very Interesting" photographs (1373): 72%
- Number of false positive detections: 392
- Total number of detections: 1380
- Error rate: 28%

Threshold: 3
- Known "Very Interesting" photographs detected: 609
- Percentage of total number of known "Very Interesting" photographs (1373): 44%
- Number of false positive detections: 183
- Total number of detections: 792
- Error rate: 23%

Threshold: 4
- Known "Very Interesting" photographs detected: 355
- Percentage of total number of known "Very Interesting" photographs (1373): 26%
- Number of false positive detections: 59
- Total number of detections: 414
- Error rate: 14%

Threshold: 5
- Known "Very Interesting" photographs detected: 247
- Percentage of total number of known "Very Interesting" photographs (1373): 18%
- Number of false positive detections: 8
- Total number of detections: 255
- Error rate: 3%

These results are promising, because they demonstrate the ability of the trained neural networks to produce accurate detections of "Very Interesting" photographs when used in conjunction with a threshold. However, many of the "Very Interesting" photographs are misclassified as "Somewhat Interesting" or "Not Interesting" when the threshold is increased. These results are greatly improved by merging the "Somewhat Interesting" category with the "Very Interesting" category. Thus, the output of the neural network can be simplified to classifying between images that are "Interesting" or "Not Interesting." Results reflecting this merge as as follows:

Threshold: 1

- Known "Interesting" photographs detected: 1859
- Percentage of total number of known "Interesting" photographs (1930): 96%
- Number of false positive detections: 190
- Total number of detections: 2049
- Error rate: 9%

Threshold: 2

- Known "Interesting" photographs detected: 1604
- Percentage of total number of known "Interesting" photographs (1930): 83%
- Number of false positive detections: 162
- Total number of detections: 1766
- Error rate: 9%

Threshold: 3

- Known "Interesting" photographs detected: 1211
- Percentage of total number of known "Interesting" photographs (1930): 63%
- Number of false positive detections: 98
- Total number of detections: 1309
- Error rate: 7%

Threshold: 4

- Known "Interesting" photographs detected: 740
- Percentage of total number of known "Interesting" photographs (1930): 38%
- Number of false positive detections: 34
- Total number of detections: 774
- Error rate: 4%

Threshold: 5

- Known "Interesting" photographs detected: 409
- Percentage of total number of known "Interesting" photographs (1930): 21%
- Number of false positive detections: 9
- Total number of detections: 418
- Error rate: 2%

These results suggest that a given image's similarity to an image in the known "Interesting" group can predict its own membership to that group. This similarity is measured in terms of the counter-propagation network performing a closest-match search of the image's representative metadata to the input patterns provided during the training phase.

Subjective Experiments

The final experiments attempt to apply the trained CPN from the previous learning phase to image data sets that are not associated with quantifications of popularity. The first set of images is a random sampling of 250 of the most recent images uploaded to the Flickr web site at the time of the experiment. The second set of images is a sampling of 2912 photographs from the author's personal collection of digital photographs.

Flickr.com test set results

Input data is generated for the 250 Flickr test images in similar fashion to the training and verification test sets from the previous experiments. With a threshold of 5 (that is, all five networks classify the image as "Interesting"), only 14 images are classified as "Interesting". Figure 16 contains the 14 images selected.

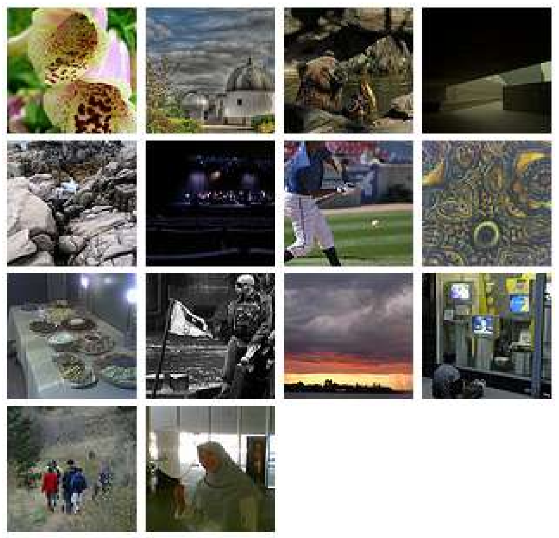
Figure 16: The 14 images classified as "Interesting" by all five counter-propagation neural networks. To see the large version of these images, go to the following URL:
http://oranchak.com/paper/flickr-5/int-5-large.html

With a threshold of 4, more photographs are classified as "Interesting". The total for this case is 57 photographs. They are displayed in Figure 17.

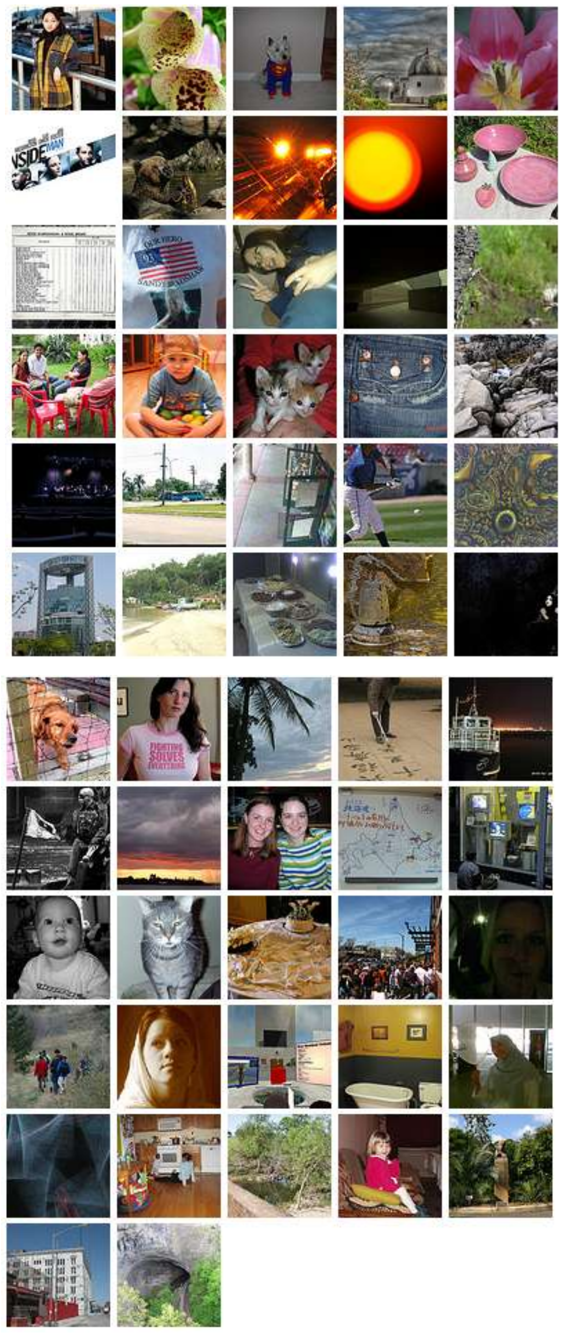

Figure 17: The 57 images classified as "Interesting" by at least four out of five of the counter-propagation neural networks. To see the large version of these images, go to the following URL:
http://oranchak.com/paper/flickr-4/int-4-large.html

Was this experiment successful? The question is very subjective and difficult to measure. In this author's opinion, most of the photographs in both threshold groups are indeed interesting. However, only a smaller number of the photographs seem to share the consistently high subjective qualities that the top ranked photographs on Flickr.com exhibit. A better measure would be to survey user reaction to test sets of photographs to measure their levels of interest in the photographs. More discussion on this topic is found in the Discussion section.

Personal photograph test set results

Input data is generated for the 2912 test images from the author's collection of digital photographs in similar fashion to the training and verification test sets from the previous experiments. With a threshold of 5 (that is, all five networks classify the image as "Interesting"), only 98 images are classified as "Interesting". Too numerous to include in this paper, they can be seen at the following URL:

http://oranchak.com/paper/personal-5/int-5.html

To this author, there is no clear majority of interesting photographs in this result. The networks have indeed selected some very interesting photographs, but too many of the photographs are ordinary family photographs with little distinguishing photographic quality. More experimentation is required to determine why the networks have more success classifying interesting images that originate from the Flickr site.

VI. DISCUSSION

Results herein indicate a high degree of success in training neural networks with the counter-propagation algorithm to predict the popularity potential of arbitrary images from the Flickr web site. A great deal of further refinement via expanded experimentation is required to improve the technique. One limitation in the current technique is that despite good accuracy on predictive capability against Flickr images with known and unknown popularity rankings, the trained neural networks do not perform well against arbitrary images from the author's personal photograph collection. One potential problem is that the input images were resampled down to the 10x10 and 20x20 pixel data sizes without preserving the aspect ratio on the images. Flickr automatically crops all of their images during resizing to preserve aspect ratio. A similar process must be applied to non-Flickr data sets to have a consistent comparison. Additionally, the non-Flickr input data used for histogram and texture analyses were not resized to the same relative scale of the medium-sized photographs obtained from Flickr. Possibly, this difference of scale may help contribute to poor network accuracy. Finally, to verify the usage of the neural network, the winner-take-all nodes of the Kohonen layers of the neural networks should be linkable back to the images they are representing. In this way, a classified image can be compared to the training images that contribute to its classification. Network parameters can be adjusted based on this comparison to tune the ability of the CPN to achieve the best match to training images. A user interface that prompts the user with such a comparison and accepts the user's "vote" about agreement on the match can significantly improve the network performance.

Additional improvements to the CPN approach in general for this problem include the following:

- Create input vectors based on texture analysis of the gray scale version of the original image data.
- Increase the height/width of the raw pixel data used as input

- Perform intelligent anti-aliasing of pixel data to preserve as much image data as possible while resizing/downsampling.
- Use a hybrid network to improve accuracy. For example, a tuning network could be attached to each of the outputs of the five counter-propagation networks to improve the classification of interesting images.
- Split up the texture input data into multiple networks to try to isolate the dependencies in the data.
- Include more types of image analysis metadata as input.
- Explore the use of other node types as replacements for the winner-take-all nodes in the Kohonen layer. For example, Gaussian nodes may provide more probabilistic information related to how well an pattern matches the network's memory of trained data. For example, Gaussian nodes may provide the ability to detect an ordered list of best matches, rather than limiting the match to exactly one image. The Euclidian distances related to multiple Gaussian nodes may provide more information about how strongly an input pattern matches the training data.
- Alter the number of output nodes to vary the range of values to which the input patterns are mapped.

Further, more exploration is required to determine the effect on CPN training using using scale- or position-dependent input data versus scale- or position-independent input data. These kinds of data can result in misclassification by CPN because a simple change in the input data can drastically change the comparison to other image data in the networks. For example, raw pixel data is position and scale dependent. A one-pixel offset in an image can completely change its computed distance from other image patterns in the CPN. Further research is needed to improve the technique of sampling portions of the input image to account for these effects.

VII. BIBLIOGRAPHY

[1] M. S. Bartlett, P. A. Viola, T. J. Sejnowski, B. A. Golomb, J. Larsen, J. C. Hager, and P. Ekman, "Classifying Facial Action", Advances in Neural Information Processing Systems 8, D. Touretzky, M. Mozer, and M. Hasselmo (Eds.), MIT Press, Cambridge, MA, 1996. p. 823-829.

[2] H. A. Rowley, S. Baluja, and T. Kanade, "Human Face Detection in Visual Scenes", in Proc. PAMI 98, 1998.

[3] M. Gallagher, and P. Deacon, "Neural Networks and the Classification of Mineralogical Samples Using X-Ray Spectra", In L. Wang et. al., editors, Proc. International Conference on Neural Information Processing (ICONIP'02), pp 2683-2687, 2002. IEEE Press, Piscataway, NJ.

[4] J. H. P. Burrill, "Texture Mapping of Neurological Magnetic Resonance Images", [Online document] 2003, Available at : http://www.burrill.demon.co.uk/meddoc/tmnmri.html

[5] M. Sonka, "Texture: Statistical texture description
", [Online document] June 1999, Available at: http://www.icaen.uiowa.edu/~dip/LECTURE/Texture1.html

[6] M. Halllbey, "Gray Level Co-Occurrence Matrix Tutorial", [Online document] 2002, Available at: http://www.fp.ucalgary.ca/mhallbey/tutorial.htm

[7] A. Hoenselaar, "JavaNNS: Java Neural Network Simulator", [Online document] 2006, Available at: http://www-ra.informatik.uni-tuebingen.de/software/JavaNNS/welcome_e.html

[8] P. Marrone, "Joone - Java Object Oriented Neural Engine", [Online document] 2004, Available at: http://www.jooneworld.com/