

# Fuzzy ART for Relatively Fast Unsupervised Image Color Quantization

Nicholas Shorter and Takis Kasparis

University of Central Florida  
School of Electrical Engineering and Computer Science, Orlando, FL, USA  
nshorter@mail.ucf.edu, kasparis@ucf.edu

**Abstract.** The use of Fuzzy Adaptive Resonance Theory (FA) is explored for the unsupervised color quantization of a color image. The red, green and blue color component values of a given color image are passed as input instances into FA which then groups similar colors into the same class. The average of all of the colors in a given class then replaces the pixel values whose original colors belonged to that class. The FA unsupervised clustering is capable of realizing color quantization with competitive accuracy and arguably low computation time.

**Keywords:** Image Color Quantization, Fuzzy ART, Clustering, Unsupervised.

## 1 Introduction

Color quantization (CQ) is the process of reducing the number of colors in a given image by replacing a group of similar colors with a single color. The replacement should be such that the color quantized image looks as similar as possible to the original image. Color quantization is often used as a preprocessor to various digital image processing applications such as segmentation. The JSEG [1] and B-JSEG [2] algorithms both rely on image CQ preprocessing. We intend to segment high resolution aerial images (4000x3000 pixels at 15cm pixel resolution) and therefore needed a computationally efficient color quantizing technique. Because of Fuzzy Adaptive Resonance Theory's (Fuzzy ART or FA) [3] ability to realize one shot fast learning and because of its competitive unsupervised learning performance; it is chosen as the clustering algorithm to group the original image's color palette into a smaller subset of colors.

## 2 Problem Formulation

### 2.1 Fuzzy ART Parameters

Carpenter et al's FA has three user defined parameters: the vigilance parameter  $\rho \in [0, 1]$ , the learning rate  $\beta \in [0, 1]$ , and the choice parameter  $\alpha > 0$ . For fast learning, the choice parameter is chosen close to 0 (actually set to  $\alpha = 0.0001$ )

and the learning rate is set to 1 ( $\beta = 1$ ). These settings therefore correspond to one shot stable learning where no weight change or search occurs after each item of an input set is presented just once.

The vigilance parameter controls how resilient FA is to accepting new input instances to existing pattern templates. Therefore, the higher (or closer to 1) the vigilance parameter is set, the more categories (or quantized colors) are created. The lower (or closer to 0) the vigilance parameter is set, the less categories (or quantized colors) are created.

## 2.2 Input Representation and Preprocessing

An input image is defined as matrix  $\bar{\mathbf{I}}_{\text{in}}$ ,  $i \in \{1, 2, \dots, R\}$ ,  $j \in \{1, 2, \dots, C\}$ , with  $R \times C$  elements. Each element in the matrix  $\bar{\mathbf{I}}_{\text{in}}$  is a 3 dimensional vector corresponding to a given pixel's color RGB components:

$$\bar{\mathbf{I}}_{\text{in}}(i, j) = \{I_{in,r}(i, j), I_{in,g}(i, j), I_{in,b}(i, j)\} \quad (1)$$

Assume the input image is 24 bits per pixel. The inputs are then presented to FA as an array of colors:

$$\bar{\mathbf{I}}_{\text{ary}}(k) = \bar{\mathbf{I}}_{\text{in}}(i \cdot j) \text{ where } k = \{1, 2, \dots, R \times C\} \quad (2)$$

The input image is read into memory via the cimg header file [4].

In [5], Moore describes a problem in FA where a large number of inputs erode the norm of weight vectors causing FA to create more categories than necessary to encode a given pattern (known as category proliferation). Carpenter et. al in [3] note that normalizing the input by compliment encoding it can enable FA to avoid the category proliferation problem. Assuming 8 bits, each color component input is scaled to exist between 0 and 1:

$$a_r(k) = I_{ary,r}(k)/255 \quad (3)$$

Equation (3) shows how the red color component is scaled. Similar equations can be derived for the blue and green color scalings. The complement of 'a' is calculated as follows:

$$a^c(k) = 1 - a(k) \quad (4)$$

The compliment encoding of the input then is formulated as follows:

$$\bar{\mathbf{X}}(k) = \{a_r(k), a_g(k), a_b(k), a_r^c(k), a_g^c(k), a_b^c(k)\} \quad (5)$$

## 2.3 Fuzzy ART Classification

The inputs (array of colors) are presented to FA which then clusters similar colors into the same class. Therefore groups of pixels are now assigned membership to a given class:

$$\bar{\mathbf{X}}^p(k) = \left\{ a_r^p(k), a_g^p(k), a_b^p(k), a_r^{p,c}(k), a_g^{p,c}(k), a_b^{p,c}(k) \right\} \quad (6)$$

$$p \in \{1, 2, \dots, C_l\}$$

where  $C_l$  is the total number of classes detected by FA and  $p$  are the pixel class labels.

The color components of all of the pixels belonging to a single class are averaged. Then for all of those pixels in that class, their colors are replaced by the class color component average:

$$A_r^p = \frac{1}{N_p} \cdot \sum_p I_{in,r}^p(i, j) \quad (7)$$

Equation (7) is the class average for the red color component. Similar equations can be derived for the blue and green color class averages. Note that  $N_p$  is the total number of points in a given class. The output,  $O(i,j)$ , for a given pixel located at  $(i,j)$  and belonging to class  $p$  is therefore the red, blue and green class color component averages:

$$\bar{\mathbf{O}}(i, j) = \{A_r^p, A_g^p, A_b^p\} \quad (8)$$

### 3 Experiments

#### 3.1 Performance Metrics

Several metrics were devised to evaluate the performance of FA's ability to color quantize a given input image. The algorithm's execution time was noted. Execution time is obviously relative to the machine which runs the algorithm. Our tests were benchmarked on an AMD 3700+ Sandiego Single Core Processor – 2.2 GHz. The computer also had 2 gigabytes of DDR400 (PC3200) ram with 2-3-2-5 timings.

In order to evaluate the quality of the color quantized image, the root mean squared error (RMSE) between the original color image and the color quantized image is used. The RMSE is also used in both [6] and [7] where they have benchmarked their algorithms on the Lenna and Mandrill images as well. The MSE for the red color component is calculated as follows:

$$MSE_r = \frac{1}{R \cdot C} \cdot \sum_{i=1}^R \sum_{j=1}^C (O_r(i, j) - I_{in,r}(i, j))^2 \quad (9)$$

Substituting the Output term for the color class average:

$$MSE_r = \frac{1}{R \cdot C} \cdot \sum_{i=1}^R \sum_{j=1}^C (A_r^p - I_{in,r}(i, j))^2 \quad (10)$$

Similar equations can be written for the MSE's for green and blue colors (denoted  $MSE_g$  and  $MSE_b$  respectively). The total RMSE then is the average of the three color component mean square errors:

$$RMSE = \sqrt{\frac{1}{3} \cdot (MSE_r + MSE_g + MSE_b)} \quad (11)$$

### 3.2 Experimental Results

Several experiments were performed to benchmark FA's clustering ability. First, three sets of 9 experiments were conducted for the entire range of the vigilance parameter. In the first set,  $\beta = 1$  and  $\alpha = 0.0001$  for one shot stable learning. Fuzzy ART was only allowed to execute for a single epoch (the Input was only presented to FA once). In the second set, the learning rate was empirically set to  $\beta = 0.8$  and the choice parameter set to  $\alpha = 0.1$ . These values were found to give the best results for when FA was allowed to execute for three epochs. Finally, the learning rate was set to  $\beta = 0.8$  and the choice parameter set to  $\alpha = 0.01$ . Now FA was allowed to run for as many epochs until it converged. Convergence for FA was defined when two conditions were met. The first condition was that no new categories were created during the current and last epoch. The second condition was that the magnitudes of the weight changes for all dimensions were less than a predefined value:

$$\forall l, \forall p \Rightarrow |w_{l,p}^{new} - w_{l,p}^{old}| \leq 0.0001 \quad (12)$$

where  $l$  is the dimension index and its range is as follows:  $l \in \{1, 2, \dots, 6\}$ . The  $p$  is the class index and its range is as follows:  $p \in \{1, 2, \dots, C_j\}$ . These three sets of 9 experiments were conducted on two images: the Lenna image (labeled as '-L') and the Mandrill image (labeled as '-M') with results shown for both images in [Fig. 1].

Then, six additional sets of experiments (testing 3 methods on both the Lenna and Mandrill images) were performed using the one shot stable learning parameters. The vigilance parameters were fine tuned so that the color quantized image had the following number of colors: 16, 32, 64, 128, 256 and 512.

First, as described in section 2.3, all R,G,B color components in a single class (as clustered by FA) were averaged and then the average replaced the pixels belonging to that class in the output image.

The vigilance parameters (Vig.), the RMSE, and the execution time in seconds (Sec) for the corresponding number of colors (Col) for both the Lenna image and the Mandrill image are presented in [Table 1] (shown on page 8).

Second, instead of using the average, we experimented with representing a given class in the output image with the median of the class' spectral components. We sorted

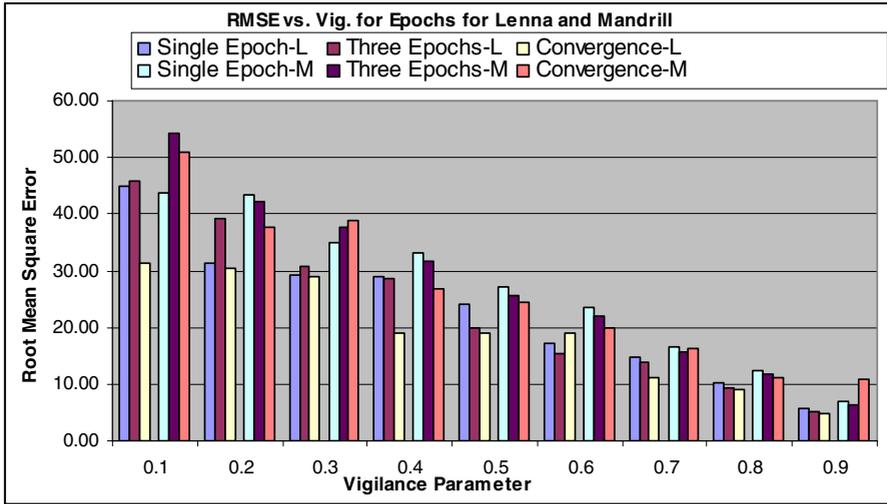


Fig. 1. RMSE for Different Epochs for Lenna

each of the R, G and B spectral components of pixels in a single class in ascending order and then found the median of each of those pixel spectral components for each class:

$$M_r^p = \text{Median} \left\{ \bar{\mathbf{I}}_{in,r}^p(i, j) \right\} \quad (13)$$

where  $M_r^p$  is the median of all the red components of all pixels in class p. Similar equations can be written for the median of all blue and green pixels in class p. Then all pixels belonging to class p are replaced in the output image with the median of their class spectral components:

$$\bar{\mathbf{O}}(i, j) = \{M_r^p, M_g^p, M_b^p\} \quad (14)$$

Third, a trimmed mean median was used to represent the pixels of a given class in the output image. First, all of the spectral components (RGB colors) for all pixels in a given class were sorted:

$$\bar{\mathbf{S}}_r^p = \text{sort} \left( \bar{\mathbf{I}}_{in,r}^p(i, j) \right) \quad (15)$$

Then, the mean was calculated for only samples from  $N_p/3$  to  $2 \cdot N_p/3$  for each color component in a given class p:

$$TA_r^p = \frac{1}{\binom{N_p}{3}} \cdot \sum_{k=N_p/3}^{2 \cdot N_p/3} S_r^p(k) \tag{16}$$

Equations similar to (15) and (16) can be written for the green and blue colors. Each output pixel is then replaced by the class' trimmed mean median:

$$\bar{O}(i, j) = \{TA_r^p, TA_g^p, TA_b^p\} \tag{17}$$

All of the different output methods (average, median of each color, and trimmed average) are all compared in terms of their RMSE for the various colors tested for the Lenna ('-L') and Mandrill ('-M') images in [Fig. 2].

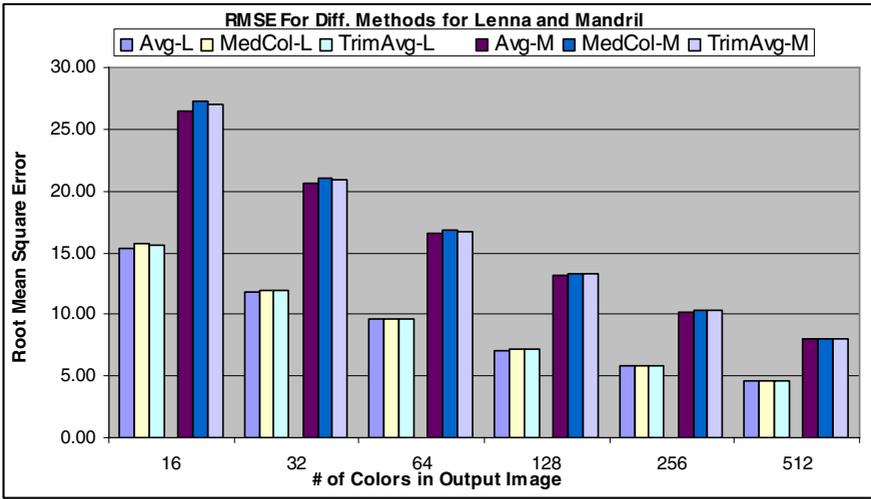


Fig. 2. RSME for Diff. Methods for Lenna



Fig. 3. Lenna as 64 Colors with FA CQ

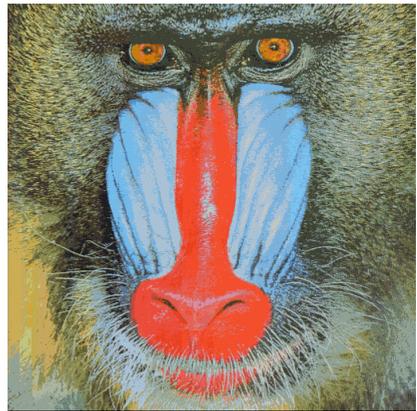


Fig. 4. Mandrill as 64 Colors with CQ

The 64 color quantized Lenna and Mandrill images (RMSE listed in [Table 1]) are presented as [Fig. 3] and [Fig. 4] respectively. Both of these figures were generated by substituting the output pixel with the class average. The original version of the standard Lenna image exists as a 512x512 pixel image with 148,279 colors while the Mandrill image is 512x512 pixels with 230,427 colors.

Finally, two additional sets of experiments were conducted where the FA color quantization algorithm was tested on two real aerial images depicting commercial and residential buildings existent in Fairfield, Australia. The aerial images have a 15 cm/pixel spacing resolution.

For these two images, only the average of each of the R,G,B spectral components of inputs belonging to the same class, as described in section 2.3, were used to represent the color clusters in the output image. The results for Scene 1 (partially depicted in [Fig. 5]) and Scene 2 (partially depicted in [Fig. 6]) are shown in [Table 2].

**Table 1.** Results for Average Each Color

Average of Each Color in Class						
Col	Lenna Image			Mandrill Image		
	Vig	RMSE	Sec	Vig	RMSE	Sec
16	0.6257	15.36	3	0.49	26.46	3.1
32	0.72	11.83	4.2	0.6085	20.59	4.2
64	0.795	9.58	6.9	0.705	16.52	7.6
128	0.849	7.09	12	0.775	13.15	11
256	0.886	5.80	22	0.83	10.22	23
512	0.9152	4.59	40	0.8719	7.95	42

**Table 2.** Results for Scenes 1 and 2

Average of Each Color in Class						
Col	Scene 1			Scene 2		
	Vig	RMSE	Sec	Vig	RMSE	Sec
16	0.5	22.24	34	0.5	20.66	32
32	0.61	15.92	51	0.64	12.33	50
64	0.705	11.44	89	0.73	10.16	85
128	0.76	9.80	142	0.79	9.02	139
256	0.825	8.35	295	0.84	7.41	259
512	0.87	6.81	583	0.885	5.86	576



**Fig. 5.** Scene 1 Cropped



**Fig. 6.** Scene 2 Cropped



**Fig. 7.** Scene 1 Quantized to 64 Colors **Fig. 8.** Scene 2 Quantized to 64 Colors Cropped

Cropped sections of the 64 color quantized Scene 1 (1510x1973 pixels with 698,843 colors) and Scene 2 (1595x1878 with 519,513 colors) are shown in [Fig. 7] and [Fig. 8] respectively. Both pictures were about 8.5 megabytes in size. The cropped sections shown were selected where there existed a rather large amount of colors so that the color quantization errors were somewhat visible.

This algorithm's executable is publicly available at [8].

## 4 Conclusions

Because the original Mandrill image has about 80,000 more colors than the Lenna image, it makes sense that in [Fig. 1] all of the RMSE values are higher for the Mandrill than that of the Lenna image for the listed, quantized colors.

Observe how letting FA execute for 1 to as many epochs as needed for convergence for both the Lenna and the Mandrill does not result in a significant reduction in the RMSE for vigilance values of 0.5 (which in most cases correspond 16 or more resultant quantized colors) and higher (see [Fig. 1]). Therefore, we recommend the use of one shot stable learning ( $\beta = 1$ ,  $\alpha = 0.0001$ , and only one epoch or input list presentation) for FA input parameters as the execution time is significantly lower and the RMSE is on par with that of letting the algorithm run for additional epochs.

We have experimented with several different methods for representing a given cluster of colors existent in the input image in the output image. These methods are as follows: averaging all the colors in a given class; taking the median of the colors in a

given class; and taking a trimmed mean median of all the colors in a given class. It was observed that simply averaging all the colors in a given class and replacing the pixels in that class with those color averages worked the best, yielding not only the lowest RMSE, for the tested Lenna and Mandrill images as shown in [Fig. 2] but the lowest execution time as well (as opposed to using the other two median based output representation methods).

In addition to testing the algorithm on the Lenna and Mandrill images, tests for various color quantizations were also done on real images captured over Fairfield, Australia. The proposed implementation produced acceptable color quantized versions of the images with an arguably low execution time and RMSE.

The use of FA therefore has several advantages for CQ. One of which is its low execution time (on the order of less than a minute for both the Lenna and Mandrill images when the number of colors desired is 512 or less, see [Table 1]) and on the order of minutes for the larger aerial images (see [Table 2]). The RMSE yielded between the original and color quantized image, as shown in [Fig. 2], is competitive with that of other methods in the literature [6], [7]. One unfortunate disadvantage of using FA for CQ is one cannot directly specify the number of colors that results in the color quantized image.

## 5 Future Work

One possible future direction in which to take this work would be to explore the use of additional features to enable the FA clustering to account for the pixel's location and context in the image in addition to the pixel's RGB value. This could possibly further improve the algorithm's performance.

## Acknowledgments

This research effort has been funded by a grant from Harris Corporation. The authors would like to thank the anonymous reviewers for their critiques. The authors would also like to thank Dr. Simone Clode and Dr. Franz Rottensteiner for donating the Fairfield, Dataset which was used in this research effort.

## References

1. Deng, Y., Manjunath, B.S.: Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(8), 800–810 (2001)
2. Wang, Y.-G., Yang, J., Chang, Y.-C.: Color-texture image segmentation by integrating directional operators into JSEG method. *Pattern Recognition Letters* 27 (2006)
3. Carpenter, G.A., Grossberg, S., Rosen, D.B.: *Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System* (1991)
4. CImg Header File, <http://cimg.sourceforge.net/>
5. Moore, B.: ART1 and Pattern Clustering. In: *Proceedings of the 1998 Connectionist Models Summer School*, pp. 174–185. Morgan Kaufmann Publishers, San Mateo (1989)

6. Ashutosh, D., Bose, N.S.C., Kandula, P., Kalra, P.K.: Modified Forward Only Counterpropagation Network (MFOCPN) for Improved Color Quantization by Entropy based Sub-clustering. In: International Joint Conference on Neural Networks, IJCNN 2007, pp. 1865–1870 (2007)
7. El-Mihoub, T., Nolle, L., Schaefer, G., Nakashima, T., Hopgood, A.: A Self-Adaptive Hybrid Genetic Algorithm for Color Clustering. In: IEEE International Conference on Systems, Man and Cybernetics, SMC 2006, October 8-11, vol. 4, pp. 3158–3163 (2006)
8. Nicholas Shorter's research site, <http://www.nshorter.com/>