

# Realtime Recoloring Objects using Artificial Neural Networks through a Cellphone

Martín Montes Rivera<sup>1</sup>, Alejandro Padilla<sup>2</sup>, Juana Canul<sup>3</sup>, Julio Ponce<sup>2</sup>,  
Alberto Ochoa Zezzatti<sup>3</sup>

<sup>1</sup> Universidad Politécnica de Aguascalientes, Mexico

<sup>2</sup> Universidad Autónoma de Aguascalientes, Mexico

<sup>3</sup> Universidad Juárez Autónoma de Tabasco, Mexico

martin.montes@upa.edu.mx, apadilla2004@hotmail.com,  
juana.canul@ujat.mx, julk.ponce@gmail.com, alberto.ochoa@uacj.mx.

**Abstract.** Recoloring it is a technique for changing the color an image resulting in a different new one. Recoloration is a common photo edition operation since digital images are around every media resource and several algorithms are used for editing these pictures, nevertheless, recent digital cameras have increased enormously the quantity of pixels for producing them. This increase in the size of digital images makes difficult the recoloring operation. In order to solve the recoloring problem, there had been applied several algorithms, some algorithms directly detect the color by performing transformations on color representations to different spaces where color is easily separated but this transformation require several no linear operations. On the other hand, numerical parameters on CNNs make than this approach cannot be trained or implemented on a mobile device, more over the time required for computing an input image will made that the processed pictures be delayed continually. Considering this limitation is proposed a specific short architecture for detecting a specific color in general objects using a feedforward neural network trained with gradient descent backpropagation with variable learning rate.

**Keywords:** realtime recoloring images, artificial neural networks, gradient descent backpropagation.

## 1 Introduction

Recoloring it is a technique for changing the color or the theme of an image resulting in a different new one with almost imperceptible changes for the human eye. Recoloration is a common photo edition operation since digital images are around every media resource and several algorithms are used for editing these pictures. Nevertheless, recent digital cameras have a great quality and definition, increasing enormously the quantity of pixels for producing them. This increase in the size of digital images makes difficult the recoloring operation, especially when is required to be performed in real time, like is desired in an augmented reality device (Yan, Ren, & Cao, 2018).

Augmented Reality (AR) is an emerging technology where real elements are added with artificial elements, allowing overlapping virtual and real environments, the use of

AR is not limited to visual content, but it is its most common representation using displays, cameras and sensors for the experience. This is been applied to several areas, like games, industry, education, communication, tourism, marketing, among others (Tussyadiah, Jung, & tom Dieck, 2017).

In order to solve the recoloring problem there had been applied several algorithms, some algorithms directly detect the color by performing transformations on color representations to different spaces where color is easily separated and noise or other kind of variations like luminance are suppressed, some of this transformations include change to CIE L\*a\*b, HSV, HSM, among others, but the required operations for completely transform the image demands high computational power due to no linearities in the transformations and the required operations (Montes Rivera, Padilla Díaz, & Ponce Gallegos, Comparative between RGB and HSV color representations for color segmentation when it is applied with artificial neural networks and evolutionary algorithms, 2016).

Despite of the time required for changing a color representation in an image it is possible to determine its color only by using labeled input data with images of the color to detect like is described in (Montes Rivera, Padilla, Canul, Ponce, & Ochoa, 2018), where several colors can be classified without using any color transformation.

The detected colors are classified with specific linear equations that describe the color. Coefficients in the equations are updated depending from the color to classify these numerical parameters which are obtained using comparatively Particle Swarm Optimization (PSO) and Genetic Algorithms (GA), despite of the results in (Montes Rivera, Padilla, Canul, Ponce, & Ochoa, 2018) working with linear equations implies that colors are linearly separable but in several cases this will limit the algorithm.

Another technique that has exhibit great results recoloring pictures in Artificial Intelligence (AI) are Artificial Neural Networks, where recently its limitation as linear classifiers has been improved with several hidden layers, Graphics Processing Unit (GPU) optimization of training algorithms, regularization in training algorithms, the increasing access to huge labeled data sets and the improvement of processing hardware (Goodfellow, Bengio, & Courville, 2016).

The Work in (Levinshtein, et al., 2017) performs Realtime recoloring of hair using a convolutional neural network (CNN) for detecting the hair and then performs a color transformation in the detected hair, the performance reported is of 300 ms per photogram for mobile devices but the required time for training the large amount of data in a CNN makes than this approach cannot be trained on a mobil device, more over the time required for computing an input image will made that the processed pictures be delayed continually.

On the other hand, Artificial Neural Networks with few hidden layers would produce good results if labeled input data is correctly introduced and algorithms for training are well optimized like shown in (Goodfellow, Bengio, & Courville, 2016).

Considering this alternative is proposed a specific short architecture for detecting a specific color in general objects using a feedforward neural network trained with input data of the pixels of the objects that must be transformed, transformation is reached in real time decreasing the time for computing per photogram and maintaining quality during recoloring.

This paper is organized as follows, section 1 describes the problem and general ideas in the state of the art, section 2 describes concepts required for implementing the methodology in section 3, results are shown in section 4 and section 5 expose the conclusions of this work based on the obtained results.

## 2 Theoretical Framework

ANNs are mathematical models that was inspired by the human nervous system, they can solve complex tasks by learning numerical values that represent its synapsis and have been used since the early 50s (Engelbrecht, 2007). The very first architectures were highly limited until backpropagation allow to train hidden layers allowing to separate no linearly separable classes in a supervised training algorithm which is one of the most popular algorithms used today in commercial applications (Goodfellow, Bengio, & Courville, 2016).

Biological neurons have inputs called dendrites and a single output called axon with on or off state depending from the dendrites and its soma which is the body of the cell that classify those inputs. Synapse is the number of dendrites linked to a specific path. ANNs first architecture was proposed by McCulloch and Pits in 1943 has inputs  $x_i$  and output  $y$  and its synapse is reproduced with  $w_i$  weights, like is shown in the figure bellow and its activation function representing the soma is the shown in equation (1) (Nguyen, Prasad, Walker, & Walker, 2003):

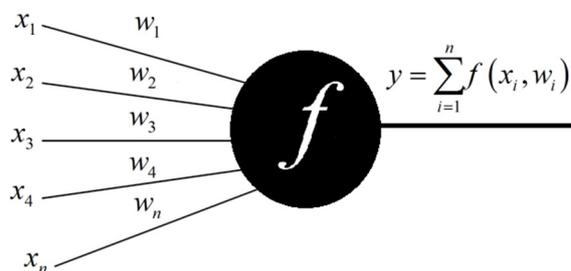


Fig. 1. McCulloch and Pits ANN.

$$F(z) = \begin{cases} 1 & z \geq \theta \\ 0 & z < \theta \end{cases} \quad (1)$$

$z$  is the total input to the neuron multiplied by  $w_i$  weights, like in equation (2):

$$z = \sum_{i=1}^n x_i w_i \quad (2)$$

Despite of McCulloch and Pits neuron was the first introduced, was hardly limited because in it is a binary neuron that could not have several layers like occurred with the first ANNs and it could separate only linear classes (Goodfellow, Bengio, & Courville, 2016).

Backpropagation Gradient Descent is the most popular learning rule that allows to train ANNs with hidden layers, and it was initially used a with a sigmoid activation function with outputs in range  $0 \leq y \leq 1$ , In this work is used a hyperbolic sigmoid activation function (3) for hidden layers since this function accelerates the convergence of backpropagation method:

$$\tan sh(z) = \frac{2}{(1+e^{-z})-1} \quad (3)$$

$$linear(z) = z \quad (4)$$

The proposed architecture is trained using Backpropagation algorithm learning rule updating weights with equations (5) and (6):

$$w_{ij} = w_{ij} + \alpha x_i \delta_j \quad (5)$$

$$w_{jk} = w_{jk} + \alpha y_j \delta_k \quad (6)$$

With  $\delta_k = \frac{\partial y_k}{\partial x_k} e_k$  and  $\delta_j = \frac{\partial y_j}{\partial x_j} \sum_{k=1}^n w_{jk} \delta_k$  with  $e_k = d_k - y_k$  where  $d_k$  is

the desired output in the  $y_k$  output and  $k$  is the index of the number of outputs,  $i$  is the index for the number of connection in the  $j$  layer of the neuron.

### 3 Methodology

The color selected for classification must be photographed in the object, then all pixels in the image  $M$  are marked as white or equal to 1, in a modified image  $M^*$  in bright of pixels, i.e. pixels take value of 1 if they belong to the color  $C$  or reduce its value for a correct classification if they do not belong, like is shown in equation (7):

$$M^*(i, j, c) = \begin{cases} 1 & M(i, j, c) \in C \\ 0.9M(i, j, c) & M(i, j, c) \notin C \end{cases} \quad (7)$$

Output vector for training is obtaining by scanning values of  $M^*$  if they are 1 then output is equal to 1 if they do not then output is equal to 0:

$$O = \begin{cases} 1 & M(i, j, c) = 1 \\ 0 & M(i, j, c) < 1 \end{cases} \quad (8)$$

The proposed architecture (Fig. 2) then is trained with the input data  $M$  and the desired output data  $O$ , using those transfer functions suggested in section 2.

ANN output is maintained as line for having a scalar value indicating a percentage of security of activation when 1 will represent 100%, since desired output is 0 or 1 depending from  $M^*$ .

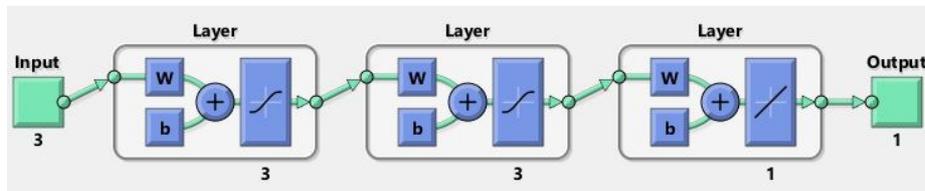


Fig. 2. Proposed Architecture for recoloring objects of a specific color.

After that colors of pixels detected by neural network are transformed by following and equation depending from its RGB values equation (7):

$$\begin{bmatrix} NR \\ NG \\ NB \end{bmatrix} = \begin{bmatrix} H(R, G, B) \\ I(R, G, B) \\ J(R, G, B) \end{bmatrix} \quad (9)$$

## 4 Results

Some of images used as input data are shown below, the images labeled by the user ( $M^*$ ) are shown in the first row of Fig. 3 and the original images ( $M$ ) are in the second row Fig. 3 as described in equation (7)



Fig. 3. Training input example for color transformation.

Then the ANN with architecture in Fig. 2 is trained using the input  $M$  in and the output  $O$  obtained from  $M^*$  using equation (8).

The training response for green color using the proposed method is shown in Fig. 4.

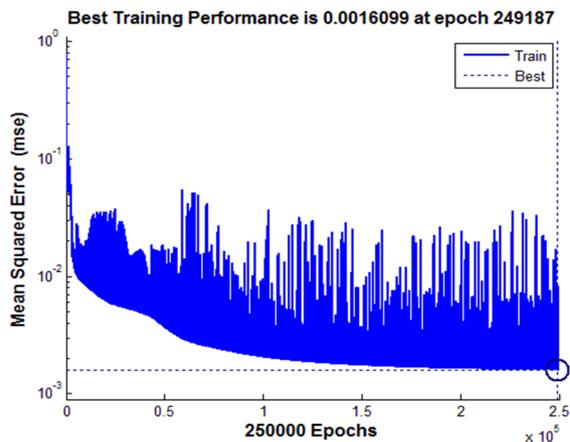


Fig. 4. Training response for ANN proposed.

The processed images presented as results are generated with Matlab™ using a PC with Core I7 6700 3.4 Ghz processor and parallel computing toolbox using NVIDIA GeForce GTX 970, the parallelizing allows to process every pixel in the images with 960x720 resolution reaching 8.1483 Frames Per Second (FPS), the images are acquired using a Samsung™ S Note 8 cellphone wireless connected to the computer using DroidCam™ app, for sending the images to the computer.

Figures from 5 to 9 shows a comparison between processed image and original transforming its color using  $R = 0.7G + R$   $B = B$ ,  $G = 0$  as functions described in equation (9), maintaining an equation structure depending from color makes possible to retain bright of colors and shadows.

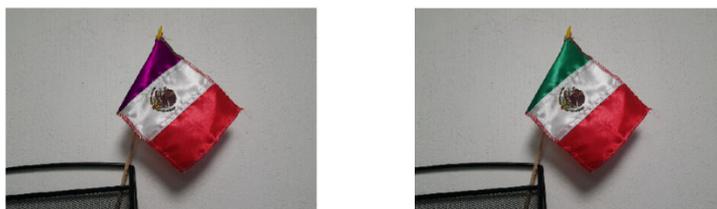


Fig. 5. Result 1 recoloring image (right processed image, left original image).

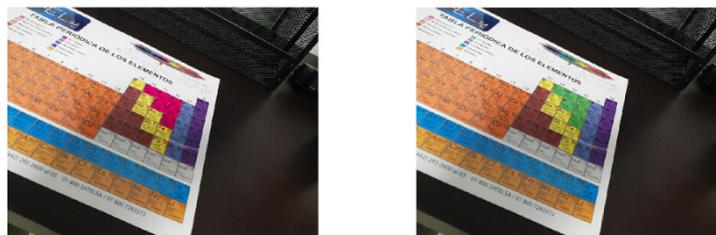


Fig. 6. Result 2 recoloring image (right processed image, left original image).

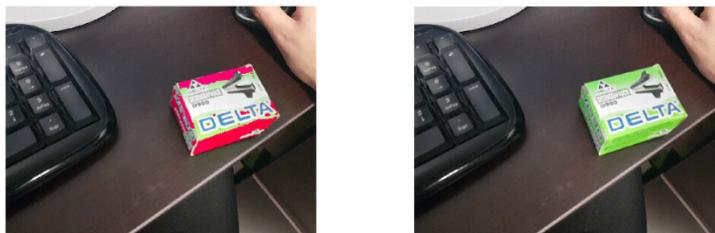


Fig. 7. Result 3 recoloring image (right processed image, left original image).



Fig. 8. Result 4 recoloring image (right processed image, left original image).



Fig. 9. Result 5 recoloring image (right processed image, left original image).

Table 1. Comparative of number of layers for recoloring images.

Work described	Type of ANN	Layers	Accuracy
(Baldassarre, Gnzalez Morín, & Rodés-Guirao, 2017)	CNN	14	80%
(Zhang, y otros, 2017)	CNN	27	Uses predicting gray for every pixel (PSNR 22.8)
(Ci, Ma, Wang, Li, & Luo, 2018)	CNN	12	Uses Mean opinion score (MOS)
This work	Feedforward ANN	2	Only training measure 1.6e-3 Mean Square Error (MSE)

This work uses a feedforward ANN with the architecture described in Fig 2. For recoloring a specific color and like is described in table 1 its comparison with other architectures is minimal making that this ANN become a good model specially when computational power is reduced and real-time is required, nevertheless some of the other architectures work with several colors in the same ANN, but require deep Convolutional Neural Networks (CNNs).

## **5 Conclusions**

In this paper is presented a recoloring technique for fast computation that runs real time recoloration with pictures acquired from a cellphone and processed using a feedforward neural network with only 2 layers, the labeled input colors are selected by the user for its change and then are trained in the neural network using gradient descent backpropagation with variable learning rate.

The recoloring task is performed with a very short neural network compared with those in the state of art detailed in table 1.

Despite of the reduced ANN architecture there are not important errors on the detection of the trained color like is shown in figures 5 to 9, however, there must be evaluated the efficiency of the algorithm comparing pictures received against pictures desired in order to measure the possible error.

The proposed technique allows the user to train a pattern color recognition by only selecting the color that must be detected, the training does not require great computational power since only 2 layers are needed, moreover, with the proposed equation (9) is possible to transform the object only in color but maintaining original bright and shadows.

Since the processing speed remains only in 8.1483 FPS for images with 960x720 resolution, a different platform from Matlab<sup>TM</sup> with lower level language must be tested, nevertheless, getting this behavior without deep neural networks like in other works, allows to perform real-time recoloration, that could be used in augmented reality devices, and even in devices with less computational power than a computer like cellphones.

### **5.1 Future Work**

Development of application in C++ interface in order to directly developing a CUDA<sup>TM</sup> kernel and perform its adaptation to the android system using OpenGL<sup>TM</sup> for parallelizing the colors detection and adapting the algorithm for processing and acquiring completely in the cellphone.

Perform a cross validation where desired output images not used during the training be taken for cross validation and efficiency evaluation.

Apply the recoloring technique here proposed in a real time application using a cellphone for recoloring objects that are difficult to perceive for colorblind people and assist them by modifying colors in Ishihara plates test, helping them to pass it or testing this situation in real environments where is required the modifying of a color for perceiving it.

## References

1. Baldassarre, F., Gonzalez Morín, D., & Rodés-Guirao, L. (2017). Deep Koalarization: Image Colorization using CNNs and Inception-Resnet-v2. arXiv preprint arXiv:1712.03400.
2. Ci, Y., Ma, X., Wang, Z., Li, H., & Luo, Z. (2018). User-Guided Deep Anime Line Art Colorization with Conditional Adversarial Networks. arXiv preprint arXiv:1808.03240.
3. Demuth, H., & Beale, M. (1998). *Neural Network Toolbox for Use with MATLAB*. Massachusetts: MathWorks Inc.
4. Engelbrecht, A. P. (2007). *Computacional Intelligence*. Sudafrica: Wiley.
5. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. <http://www.deeplearningbook.org>: MIT Press.
6. Levinshtein, A., Chang, C., Phung, E., Kezele, I., Guo, W., & Aarabi, P. (2017). Real-time deep hair matting on mobile devices. arXiv:1712.07168, 1-7.
7. Montes Rivera, M., Padilla Díaz, A., & Ponce Gallegos, J. (2016). Comparative between RGB and HSV color representations for color segmentation when it is applied with artificial neural networks and evolutionary algorithms. In D. A. M. en C. Ma. de Lourdes Sánchez Guerrero, *Avances en las Tecnologías de la Información* (pp. 611-629). Ciudad de México: ALFA-OMEGA.
8. Montes Rivera, M., Padilla, A., Canul, J., Ponce, J., & Ochoa, A. (2018). Comparative of Effectiveness When Classifying Colors Using RGB Image Representation with PSO with Time Decreasing Inertial Coefficient and GA Algorithms as Classifiers. In O. Castillo, P. Melin, & J. Kacprzyk, *Fuzzy Logic Augmentation of Neural and Optimization Algorithms. Studies in Computational Intelligence* 749. (pp. 527-546). Springer.
9. Nguyen, H. T., Prasad, N. R., Walker, C. L., & Walker, E. A. (2003). *A First Course in Fuzzy and Neural Control*. United States of America: Chapman and Hall.
10. Singh, U. P., & Jain, S. (2018). Optimization of neural network for nonlinear discrete time system using modified quaternion firefly algorithm: case study of Indian currency exchange rate prediction. *Soft Computing*, 8(2667–2681).
11. Tussyadiah, I. P., Jung, T. H., & tom Dieck, M. C. (2017). Embodiment of Wearable Augmented Reality Technology in Tourism Experiences. *Journal of Travel Research*, 1-15.
12. Valeriy Dubrovin, S. S. (2000). Neural Network Method in Plant Spectral Recognition. En R. S. Muttiah, *From Laboratory Spectroscopy to Remotely Sensed Spectra of Terrestrial of Terrestrial Ecosystems* (págs. 147-159). Kluwer Academic Publishers .
13. Vogl, T., Mangis, J., Rigler, A., Zink, W., & Alkon, D. (1988). Accelerating the convergence of the backpropagation method. *Biological Cybernetics*, 257-263.
14. Yan, Y., Ren, W., & Cao, X. (2018). Recolored Image Detection via a Deep Discriminative Model. *IEEE Transactions on Information Forensics and Security*, 14(1), 5 - 17.

*Martín Montes Rivera, Alejandro Padilla, Juana Canul, Julio Ponce, Alberto Ochoa Zezzatti*

15. Zhang, R., Zhu, J.-Y., Isola, P., Geng, X., Lin, A. S., Yu, T., & Efros, A. A. (2017). Real-time user-guided image colorization with learned deep priors. arXiv preprint arXiv:1705.02999.