Coloring Gray-Scale Image Using Artificial Neural Networks

Bekir Karlık (Haliç University, Department of Computer Engineering) and Mustafa Sarıöz (Fatih University, Computer Technology and Programming)

Main idea: using artificial neural networks for images coloring.

Method outline:

- RGB model is used in the method:

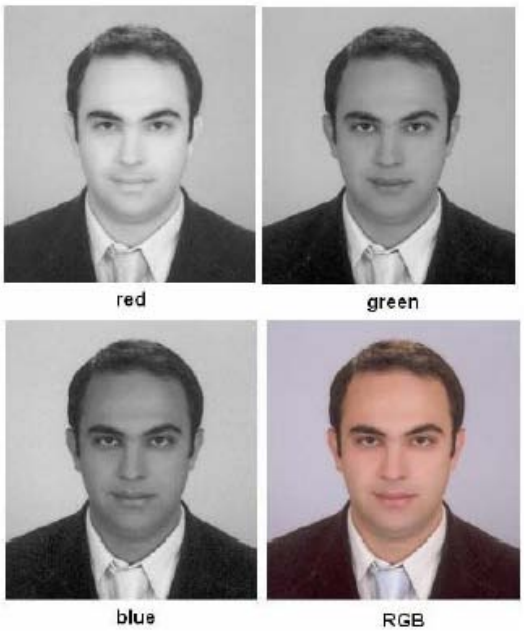


Figure 1: RGB image and components

- Six inputs are produced based on source grayscale image.

  Here are the layers of input variables (See Fig. 2):

  1. Original grayscale image.
  2. 2D matrix includes the ratio of the sum of the distances of abscissas and ordinates between the middle point to sum of middle coordinates. The size of Matrix can be varied depending to the size of training image.
     Matrix(i,j)=(abs(i-midx)+abs(j-midy))/(midx+midy);
  3. 2D blurred image matrix which is composed from computing the averages of mini matrixes which of size is given before, and writing the averages to first element of (1,1) mini matrixes.
  4. 2D matrix which helps to separate background from person in image. To compose this, following steps are done:
     - find the sum of each abscissa
     - find the sum of each ordinate
     - find values for each pixel adding together the sum of apses with the sum of ordinate
     - find average value for each pixel
     - find for each pixel value difference from the average
     - assemble these values into a matrix and send this matrix as an output
  5. 2D matrix of values which represents difference from average.

6. A new image matrix generated from the grayscale image using the sigmoid functions as sub functions.

Figure 2: Training/testing inputs

Each pixel from source grayscale image together with corresponding pixels from derived images (Fig.2) form the input for a neural network.

- Three neural networks are trained (for red, green and blue components).
- Neural network has 6 input nodes and 1 output node. Hidden layer has 10 nodes. Pureline function is used as an activation function for all layers.

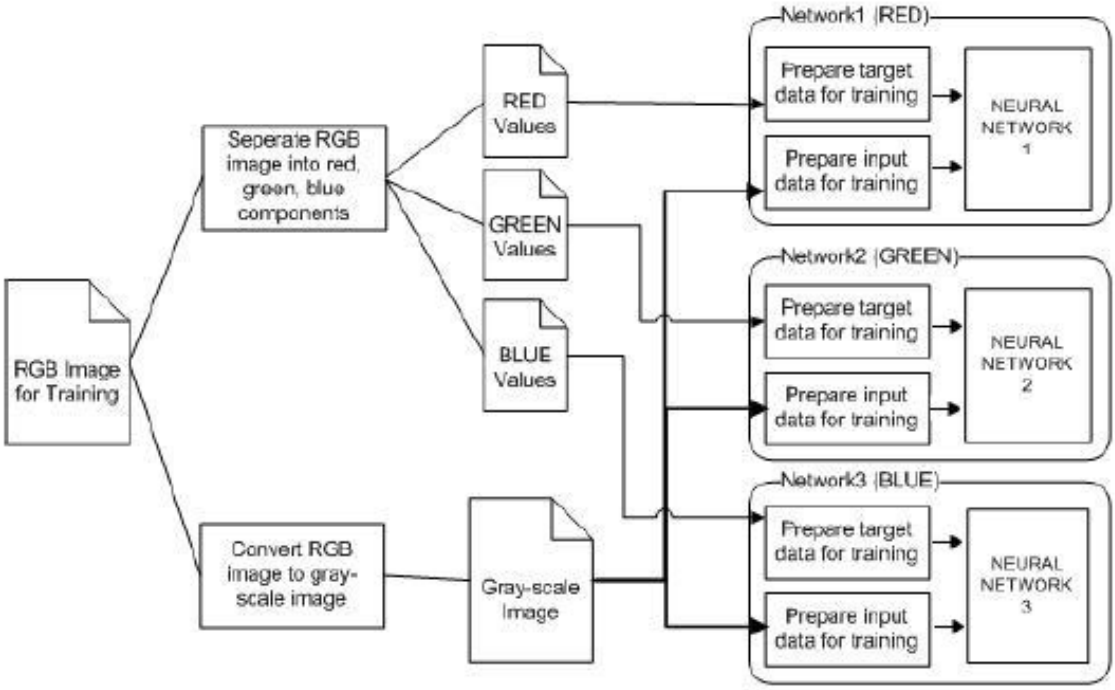


Figure 3: Pre-training and training phase

- Then trained networks are used to determine values of RGB components for each pixel of test image.

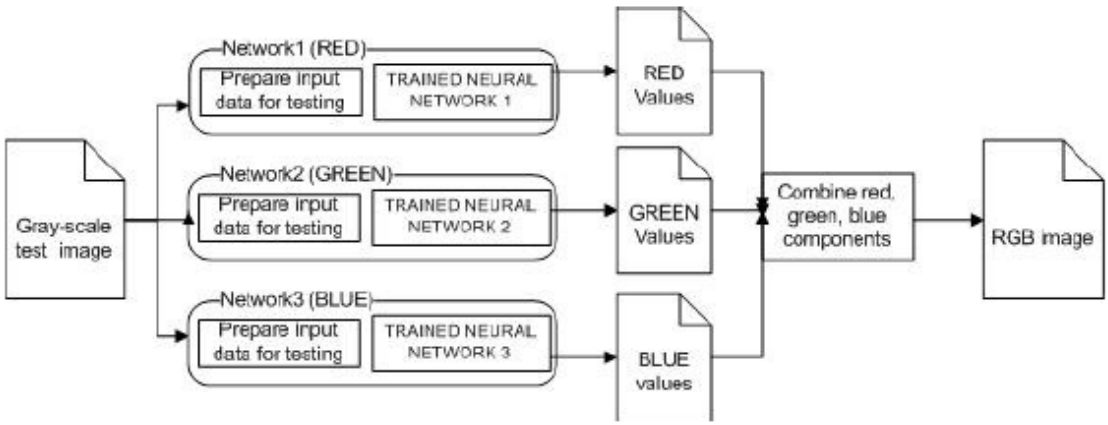
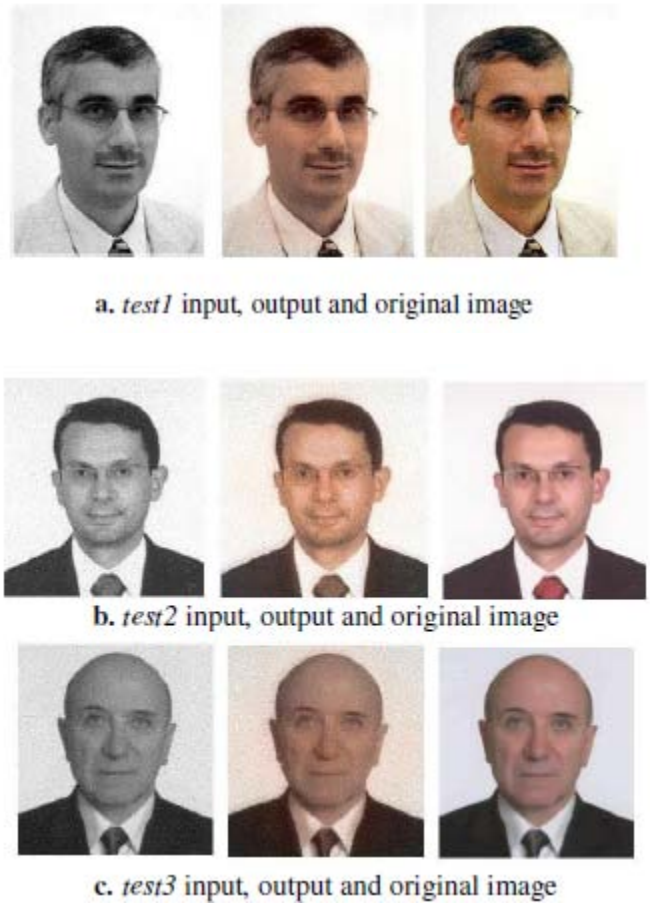


Figure 4: Testing phase

Results:



a. *test1* input, output and original image

b. *test2* input, output and original image

c. *test3* input, output and original image

Figure 5: Test inputs, outputs and original images.

It is noticed that the train data is very important for success. If we use closest images as train image, performance will be increased.