Generating Mosaic Images Based On Texture Analysis

Alaa Yaseen Taqa
alaa.taqa@gmail.com
Mosul University-Education College

Received 05/03/2013 **Accepted** 08/05/2013

تكوين صور فسيفسائية بالاعتماد على تحليل النسيج

الاء ياسين طه طاقة
كلية التربية- قسم علوم الحاسوب

خلاصة

تكوين فسيفساء الصورة (صورة فسيفسائية) هي العملية التي تقوم بتقسيم الصورة المدخلة (حاوية) الى عدد متساوي من الكتل المستطيلة (كتل الصورة)، كل منها يتم استبداله بصورة أخرى (الصورة البلاط) والتي تتطابق مع ملامح كتلة الصورة المناظرة لها. لو نظر الى فسيفساء الصورة الناتجة من مسافة فانها ستشكل صورة الادخال. أن فسيفساء الصور ليست مفهوما جديدا، ولكن هناك عدد قليل من المنشورات المتعلقة به لأنه هو مفهوم ذات اساس تجاري. جودة فسيفساء الصورة الناتجة تعتمد على حجم قاعدة البيانات وتنويع الصور. أن عنق الزجاجة لهذه المسالة هو معدل التطابق بين الصورة المدخلة فسيفساء الصورة الناتجة بالإضافة إلى الوقت المستغرق. يقترح البحث ثلاثة مناهج عمل ذكائية لانتاج فسيفساء الصور باقل وقت: المنهج الأول هو عنقدة معدلات K ومسافة مانهاتن، والمنهج الثاني هو الشبكة العصبية ذات الانتشار العكسي ومسافة مانهاتن والمنهج الثالث هو تهجين المنطق الضبابي وشكبة ايلمان العصبية.

يتم استخراج ثلاثة مجموعات من الميزات التي تستخدم من أجل العثور على أفضل تطابق بين صورة البلاط وكل كتلة صورة تناظرها داخل الصورة الحاوية. المجموعة الأولى هي الميزات الإحصائية التي يتم استخراجها من مدرج تكراري مكمم الى ٦٤ مستوى رمادي. هذه الميزات هي التباين، المعدل، التجانف، التفرطح والطاقة. والمجموعة الثانية هو مجموعة فرعية (خشونة، التناقض وتوجه) من مجموعة ميزات تامورا. في حين أن المجموعة الأخيرة من الميزات تتضمن صفة واحدة هي النسبة المئوية للخلايا الضوئية الواقعة على الحوافي داخل الصورة والتي تكتشف باستخدام مرشح كاني. الى جانب ذلك، يتم استخدام طريقتين من تصحيح الألوان لضبط الوان صورة البلاط، لتتناسب

مع ألوان كتلة الصورة المناظرة : تعتمد الطريقة الاولى على المعدل بينما تعتمد الطريقة الثانية على خصائص المدرج التكراري للالوان.

وأخيرا، فإن النتائج التجريبية تبين أن المنهج الذي يهجن المنطق الضبابي وشكبة ايلمان العصبية الافضل بين الثلاث مناهج. اذ يحتاج حوالى ١٠.٠ ثوانٍ لإنتاج فسيفساء الصور بمعدل ارتباط بين الصور المدخلة (الحاويات) وفسيفساء الصور الناتجة يساوي 0.82 باستخدام طريقة المعدل لتصحيح الالوان. من ناحية أخرى فإن هذا المنهج يحتاج حوالي ٤٢.٣٣ ثانية لإنتاج فسيفساء الصور بمعدل ارتباط يساوي ٠.٨٦ باستخدام خصائص المدرج التكراري للالوان لتصحيح الالوان.

Abstract

Photo-mosaic (mosaic-image) generation is the process of dividing an input image into equal rectangular blocks (image blocks), each of which is replaced with another image (tile image) that matches the features of a corresponding image block. When the produced photo mosaic is seen from a distance, it seemingly forms the input image.

Photo mosaic is not a new concept; however, only a few publications on this subject are available because of its commercial nature. The quality of the produced photo mosaic depends on the size of the database and the variety of images. In addition to process time, a bottleneck occurs because of the discrepancy in the match rate between the input image and the produced photo mosaic.

This research proposes three intelligence-based approaches for producing photo mosaics in less time: (1) *k*-means clustering with Manhattan distance, (2) back propagation neural network with Manhattan distance, and (3) hybrid fuzzy logic with Elman neural network.

Three groups of features are extracted and then used to find the best matching tile image for each corresponding image block within the container image. The first group comprises statistical features extracted from a 64-gray level quantized histogram. These features are variance, mean, skewness, kurtosis, and energy. The second group is a subset of Tamura features, namely, coarseness, contrast, and directionality. The last group includes a feature called edge rate, which is computed as the percentage of edge pixels within an image that can be detected using a Canny filter. In addition, two methods of color correction are used to adjust the colors of a tile image to match the colors of a corresponding image block. The first method is based on mean, whereas the second is based on histogram specification.

Finally, experimental results show that hybrid fuzzy logic with Elman neural network is the best among the three approaches used. This technique needs 10.0 seconds to produce photo mosaics, with a correlation rate of 0.82 between the container image and the produced photo mosaics, using mean-based color correction. By contrast, this approach needs 42.33 seconds to produce photo mosaics, with a correlation rate of 0.86, using histogram specification-based color correction.

Key Words: Mosaic image, Retrieval image , texture analysis , artificial techniques

1. Introduction

Art plays a very important role in computer science. This field has seen an extensive range of works created using computers as media or tools. A large number of people think that science and art are two mutually exclusive perspectives; in practice, however, scientific concepts can be used to guide artists, whereas art can be an inspiration for science. The relationship between art and science is clear, particularly in the fields of image processing [1] and computer graphics [2]. Both fields can be extended to include computer animation [3] and computer-generated imagery, which play essential roles in filmmaking [4]. Furthermore, creating tile mosaic images via computers is a recent research topic which aims to transform a raster input image into a good-quality mosaic [5].

2. Image Mosaics

Photo-mosaic generation is a technique that transforms an input image into a rectangular grid of thumbnail images [6]. It uses small photos in the same manner that a conventional graphic image uses pixels [7]. A photo mosaic has a visual content as a whole, and each of its building images also has a meaningful content [8]. A photo mosaic needs to be viewed from a distance to see the overall effect [7].

Battiato *et al*. [8] classified image mosaics into four main types: crystallization mosaic, ancient mosaic, photo mosaic, and puzzle-image mosaic. The first two types are obtained by decomposing a source image into tiles (with different colors, sizes, and rotations), and then, reconstructing the image by properly painting the tiles. As such, images under the first two classifications may be called tile mosaics. The last two types of mosaics are obtained by fitting images from a database to cover an assigned source image; thus, images under these types may be called multi-picture mosaics [9]. A different kind of mosaic, called Jigsaw Image Mosaic (JIM), is created using image tiles with arbitrary shapes to compose the final picture [10]. Another type of mosaic, called artificial mosaic, was presented by Blasi and Gallo [11]. This type of image mosaic is created based on reproducing colors of the original image and emphasizing relevant boundaries by placing tiles along edge directions.

3. Previous Works

In photo-mosaic creation literature, the works which are most closely related to our approach are various mosaicing algorithms that can be categorized by choice of tiles and restriction on their placement. Over the past decade, several photo-mosaic generation algorithms were developed.

In 2000, Robert Silvers [12] devised and patented a process called Photomosaic to generate image mosaics. His interest was commercial; therefore, details regarding his process were not much. Lai [13] pointed out that Kim [14] extended the idea of placing image tiles in a regular manner to placing them to fill a region of arbitrary shape. Tauheed [15] proposed an algorithm for creating comparatively accurate photo collages. The quantification of image similarity of Tauheed's proposed algorithm was based on color, edge, and shape features. The differentiating result of this algorithm was attained by simulating different aspects of the visual perception mechanism of human beings to obtain an aesthetically satisfying output. Blasi *et al*. [16] presented a technique to speed up the critical phase using the antipole tree data structure. This improvement allows the use of a larger database for mosaic tinning without requiring a much longer processing time. In 2010, Mikamo *et al.* [17] provided alternative methods to prevent repetitions within mosaic images while staying robust enough to work with coarse image subsets. Sah *et al.* [18] investigated the use of color adjustment and tile size variation techniques via genetic programming to improve animated photo mosaics. This investigation was able to produce an aesthetically different animation effect and presented a better mechanism for generating photo mosaics when only a limited number of tiles are available. In 2011, Lee *et al.* [19] presented a photo-mosaic smartphone application in client-server-based large-scale image databases. They proposed a best-match algorithm that exploits lower bounding property of image-PAA. They used an Android ©-based application and demonstrated its feasibility. Recently, Miller and Mould [20] proposed a system for arranging images from a database into a collage that resembles a particular target image. This collage exploits large-scale visual correspondences between the target image and the images in the database. In addition, Miller and Mould also proposed a novel color correction scheme suitable for their application. However, because these authors performed matching on the image level, they found a way to approximately isolate objects in an image using base and detail layer decomposition. Even when applied to arbitrary textures, which are not necessarily photo collages, this color correction method can create fascinating double images. This research proposes three intelligence-based approaches for producing photo mosaics in less time and high match rate between the input image and the produced photo mosaic: (1) *k*-means clustering with Manhattan distance, (2) back propagation neural network with Manhattan distance, and (3) hybrid fuzzy logic with Elman neural network.

4. Features Types

The most common image features used are color, texture, and shape. Color can be determined directly by using a color histogram [21] which shows pixel distribution of each color within the image. Statistical features can be extracted either from a color- or gray-scale histogram [22]. These features describe color and texture of an image. Furthermore, texture properties can be estimated using the Tamura [23] model, which numerically estimates coarseness, contrast, directionality, line-likeness, regularity, and roughness. The percentage of edge pixels in an image, which are detected using a Canny filter [24], can be used as an indicator of texture type. The proposed mosaic-image generator uses statistical features extracted from the gray-scale (intensity) histogram of an image, the Tamura features, and the percentage of edge pixels in the image using a Canny filter.

4.1 Statistical Features

Statistical methods analyze the spatial distribution of gray values by computing local features at each point in the image, and then deriving a set of statistics from the distribution of local features [25]. A gray-scale histogram serves as an effective representation of the color content of an image. This histogram is easy to compute and is effective in characterizing global and local distributions of intensities (colors) in an image [26]. The histogram-based approach to texture analysis depends on intensity value concentrations on all or part of an image represented as a histogram [27]. The values of a histogram depend only on individual pixel values and not on the interaction or co-occurrence among neighboring pixel values [28]. A gray-scale histogram contains very useful information about images, such as moments, which has been extensively used to describe texture. Moments produce characterizations of textures such as fine, coarse, and so on. [29]. Moments of a gray-scale histogram of an image include mean, variance, skewness, kurtosis, and energy [29][27]. Mean and variance are calculated using Equations 1 and 2, respectively [30][31], as follows:

$$mean = \frac{1}{m}\sum_{i=0}^{n} i * h(i) \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \; 1$$

$$variance = \frac{1}{m}\sum_{i=0}^{n}(i - mean)^2 * h(i) \; \dots\dots\dots\dots\dots\dots\dots\dots \; 2$$

Where i is a gray scale (level), n is a total number of bins (levels) in quantized gray-scale histogram and m is the image size in pixels.

Moreover, skewness, kurtosis, and energy are calculated using Equations 3, 4, and 5, respectively [30][31]:

$$Skewness = \frac{1}{sd^3} * \sum_{i=0}^{n} (i - mean)^3 * P(i) \text{ .... } 3$$

$$Kurtosis = \frac{1}{sd^4} * \sum_{i=0}^{n} (i - mean)^4 * P(i) - 3 \text{ ....} 4$$

$$Energy = \sum_{i=0}^{n} [P(i)]^2 \text{ .... } 5$$

Where *h* is the quantized gray-scale histogram of an image; *n* is the number of bins (levels) in the quantized gray-scale histogram; *m* is the image size in pixels; *sd* is the standard deviation of gray-scale images; and *p(i)* is the rate of each gray scale within an image.

4.2 Tamura Features

Tamura *et al.* [23] adopted the approach of devising texture features that correspond to human visual perception. These authors defined six textural features (coarseness, contrast, directionality, line-likeness, regularity, and roughness) and compared them with psychological measurements for human subjects. Deselaers [32] pointed out that Tamura *et al.* conducted experiments to test the significance of their proposed features. With respect to coarseness, contrast, and directionality, Tamura *et al.* [23] obtained very successful results. These three features are extremely significant in global descriptions of textures. The other three features, although related to the first three, do not contribute much to the effectiveness of texture description [33]. Coarseness, contrast, and directionality are expected to be separately useful in cases where texture differs only in one feature or in combinations of features for image classification and segmentation problems [34].

Coarseness provides information on the size of texture elements. The higher the coarseness value is, the rougher the texture [32]. Coarseness aims to identify the largest size at which texture exists, even where a smaller micro texture is found [34]. The essence of this method is to select a large size as the best option when a coarse texture is present even though a micro texture also exists, and to choose a small size when only a fine texture is present [23]. This procedure can be summarized in the following steps [23][32][34]:

1. Averages at each point over neighborhoods are computed, the linear size of which are powers of two (e.g., 1 × 1, 2 × 2, …, 32 × 32). The average over the neighborhood with size $2^k \times 2^k$ at the point *(x,y)* is:

$$A_k(x,y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} f(i,j)/2^{2k} \text{.................... } 6$$

where *f(i,j)* is the gray-scale value at pixel *(x, y).*

2. For each point, differences between pairs of averages corresponding to pairs of non-overlapping neighborhoods on opposite sides of the point in both horizontal and vertical orientations are found.

$$E_{k,h}(x,y) = \left|A_k(x+2^{k-1},y) - A_k(x-2^{k-1},y)\right| \text{ …………….. 7}$$

$$E_{k,v}(x,y) = \left|A_k(x,y+2^{k-1}) - A_k(x,y-2^{k-1})\right| \text{ …………….. 8}$$

3. At each point, the best size which exhibits the highest output value is saved:

$$S_{best}(x,y) = 2^k \text{ …………………………….…….. 9}$$

where *k* maximizes *E* in either direction, that is,

$$E_k = E_{max} = max(E_1, E_2, \dots, E_L) \text{ ……………….… 10}$$

4. Finally, a coarseness measure, F_{crs}, is found by computing the average of S_{best} over the picture:

$$F_{crs} = \frac{1}{m*n} \sum_i^m \sum_j^n S_{best}(i.j) \text{ ………………….…….. 11}$$

In simple terms, contrast stands for picture quality. In fact, it aims to capture the dynamic range of gray levels in an image, along with the polarization of the distribution of black and white. Therefore, contrast measure is defined as:

$$F_{con} = \sigma/(\alpha_4)^n \text{ …………………………………..…….. 12}$$

$$\alpha_4 = \mu_4/\sigma^4 \text{ ……………………..…….………………...….. 13}$$

where μ_4 is the fourth moment about the mean, σ^2 is the variance and α_4 is the kurtosis.

Tamura [23] found that a value of *n* equal to 1/4 will provide the closest agreement to human measurements. The directionality feature measures the total degree of directionality.

To compute for the horizontal and vertical derivatives, Δ_H and Δ_V are determined by convolution of the image *f (x,y)* with the following 3 × 3 operators, respectively:

-1	-1	-1
0	0	0
1	1	1

-1	0	1
1	0	1
-1	0	1

The θ values for all positions *(x,y)* are computed using Equation 14. These values are then used to find a 16-bin histogram $H_{D.}$ Directionality can be determined as the sum of second moments around each peak from valley to valley.

$$\theta = \frac{\pi}{2} + tan^{-1}\frac{\Delta_V(x,y)}{\Delta_H(x,y)} \text{ ………………………………... 14}$$

5. The Proposed Photo-Mosaic Generator

The proposed photo-mosaic system consists of two main phases: creating a database for features and generating a photo mosaic. The database for features is created through two main processes: preprocessing and feature extraction. The overall working framework of the proposed photo-mosaic generator is shown in Figure (1).

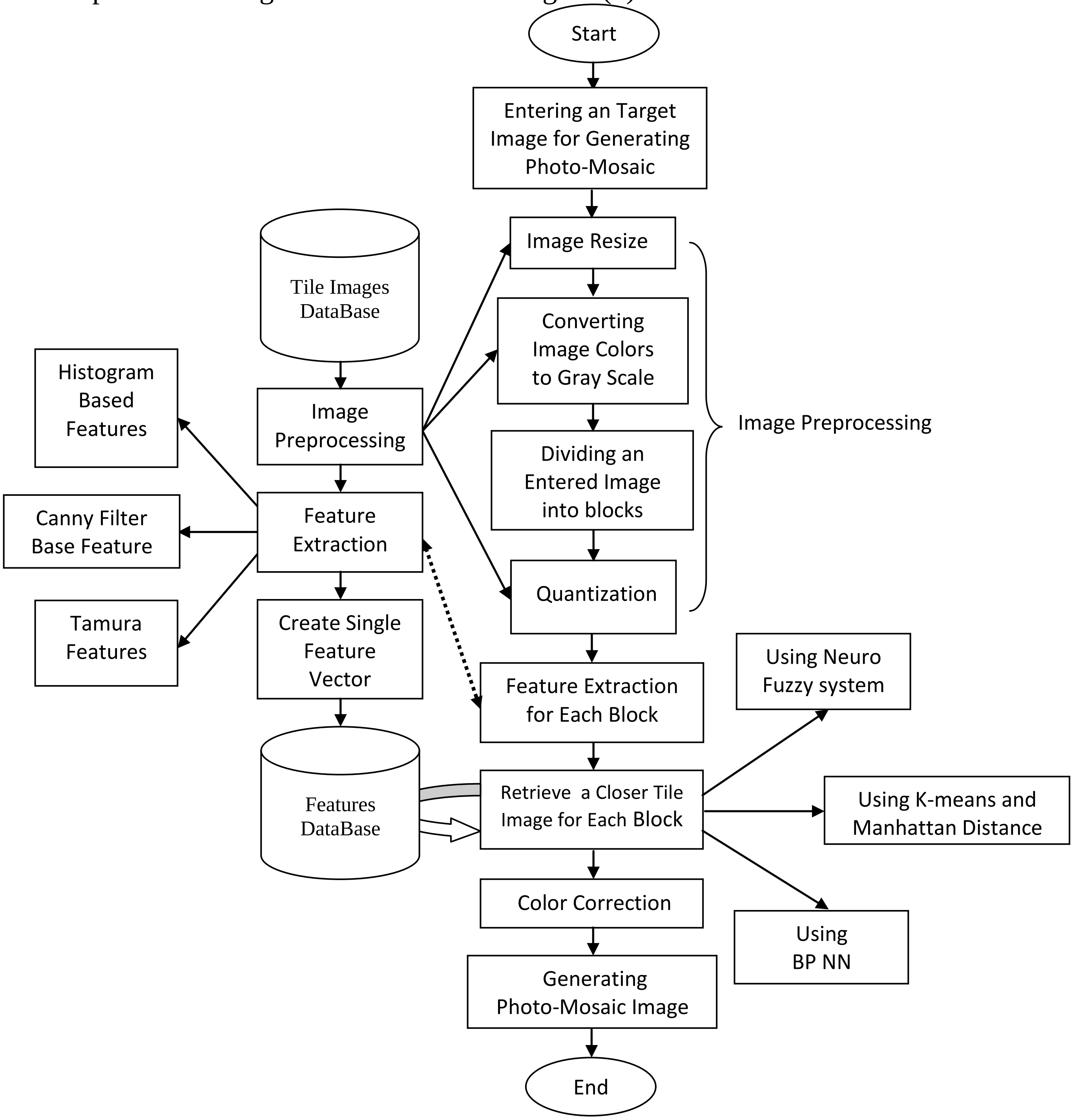

Start
Entering an Target Image for Generating Photo-Mosaic
Tile Images DataBase
Image Resize
Converting Image Colors to Gray Scale
Dividing an Entered Image into blocks
Quantization
Image Preprocessing
Histogram Based Features
Image Preprocessing
Canny Filter Base Feature
Feature Extraction
Tamura Features
Create Single Feature Vector
Features DataBase
Feature Extraction for Each Block
Using Neuro Fuzzy system
Retrieve a Closer Tile Image for Each Block
Using K-means and Manhattan Distance
Color Correction
Using BP NN
Generating Photo-Mosaic Image
End

Fig. (1) The proposed Photo-Mosaic generator

5.1 Creating a Database for Features

The only appropriate manner with regard to collecting an image database for a photo-mosaic generator is to use a large number of images drawn from a wide variety of internet sources and with various subjects (such as planes, portraits, cars, nature, art, and cities). The database used in this research contains 549 tile images which are different in sizes and subjects. The formats of the collected images are Joint Photographic Expert Group (JPEG) and bitmap (BMP). The collected tile images have been classified manually depending on their subjects. Finally, features have been extracted from each tile image in the database to form a database for features which will be used in generating photo mosaics.

5.1.1 Tile Image Preprocessing

The first step in preprocessing is size normalization, in which all tile images are resized (scaling) to a fixed size (64 × 64; 32 × 32, or 16 × 16 pixels, depending on the image block size) using a bicubic interpolation algorithm [35]. The second step is converting all color tile images into gray-scale images. The last step is quantization. During this process, the histogram is divided into bins (levels) [36]. Clausi [37] pointed out that using 64 gray levels is the best for texture features extracted from a gray-scale histogram, and thus, this work focused on the 64-gray-scale uniform quantization scheme.

5.1.2 Feature Extraction of Tile Images

Three groups of features are extracted. The first group is composed of statistical features computed based on a 64-gray-scale quantized histogram. These features are variance, mean, skewness, kurtosis, and energy.

The second group of features consists of Tamura features which are composed of six texture features (coarseness, contrast, directionality, line-likeness, regularity, and roughness). Tamura *et al.* [23] attained very successful results when they used the first three features. Accordingly, we only used these features (coarseness, contrast, and directionality) in this research.

The last group of features includes edge rate, which is computed as the percentage of edge pixels in the image. Edge-detection methods are relatively well-known and simple to implement, thus, edge density and direction have been used as quantitative methods for measuring texture in numerous applications. The number of edge pixels in several preset regions can provide insight into the overall busyness of that region [38]. Edge pixels are detected using a Canny filter. Once all the tile images in the database have their own feature vector, then the database for features will be created.

5.2 The Photo Mosaic Generator

This section describes the main steps of the photo-mosaic generator, which replaces an original image block with their corresponding best matching tile image.

5.2.1 Input Image Preprocessing

The preprocessing of input images involves four steps. The first step is size normalization which aligns the entered image (the target image converted to photo mosaic) to a fixed dimension of 640 pixels by 640 pixels using a bicubic interpolation algorithm [35]. The second step is converting input color images to gray-scale images. The third step divides an input image into blocks (image blocks), each of which is replaced by a tile image with closely similar features (color and texture). In this study, three blocks with different sizes have been selected (64 × 64, 32 × 32, and 16 × 16 pixels). Finally, color quantization [37] throughout the gray levels (intensity levels) of an image is reduced. A gray-scale image consists of 256 levels, and thus, feature extraction computation for these 256 levels is slow. To increase the speed of computation, the histogram bins (levels) of an image are reduced to 24 levels (gray scale).

5.2.2 Input Image Feature Extraction

In the proposed photo-mosaic generator, feature extraction plays a major role. Extracted features are used to find a tile image similar to an image block. When the image block is given, features are extracted in the same manner in which a tile image is extracted (Section 5.1.2). Then, a tile image resembling an image block is found based on one of the following approaches: (1) *k*-means clustering [39][40] with Manhattan distance [41][42], (2) "back propagation neural network (BPNN) [43][44] with Manhattan distance and (3) a hybrid approach combining Elman neural network [45][46][47] and fuzzy logic [48][49][50].

5.2.3 *K*-means Clustering with Manhattan Distance Approach

Abdel-Mottaleb *et al.* [51] and Wagsta_*et al.* [52] implemented *k*-means clustering algorithm as follows:

1. All feature vectors being clustered into the space are placed. These vectors represent initial group centroids. The number of clusters *nc* is chosen a priori. *nc* centers are selected by randomly picking feature vectors from the database.
2. For each feature vector in the database, the similarity measure between the feature vector and the cluster centers are computed, and the vector is assigned to the cluster in which it exhibits the largest similarity measure. As such, each feature vector is assigned to the cluster with the closest centroid.
3. New cluster centers are computed as the centroids of the clusters.
4. Steps 2 and 3 are repeated until no further change is observed in the cluster centers.

The purpose of the clustering process is to group tile images with similar color and texture features while separating dissimilar tile images.

The tile images of the database are clustered into six classes (clusters) based on their features using k-means clustering algorithm. To obtain the closest tile image to a given image block, the image block is first classified to a proper class using k-means clustering algorithm, and then, the closer tile image is chosen from that class using Equation 15 [41][42] of the Manhattan distance:

$$D_{MH}[j] = \sum_{i=1}^{N} |(F_b[i] - F_{TiC}[i][j]| \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 15$$

where $D_{MH}[j]$ is the distance between a feature vector of an image block F_b and a feature vector of tile image j in a proper class F_{TiC}, and N represents the length of the feature vector.

Figure (2) illustrates the main steps in finding the closest tile image to an image block using k-means clustering algorithm and Manhattan distance.

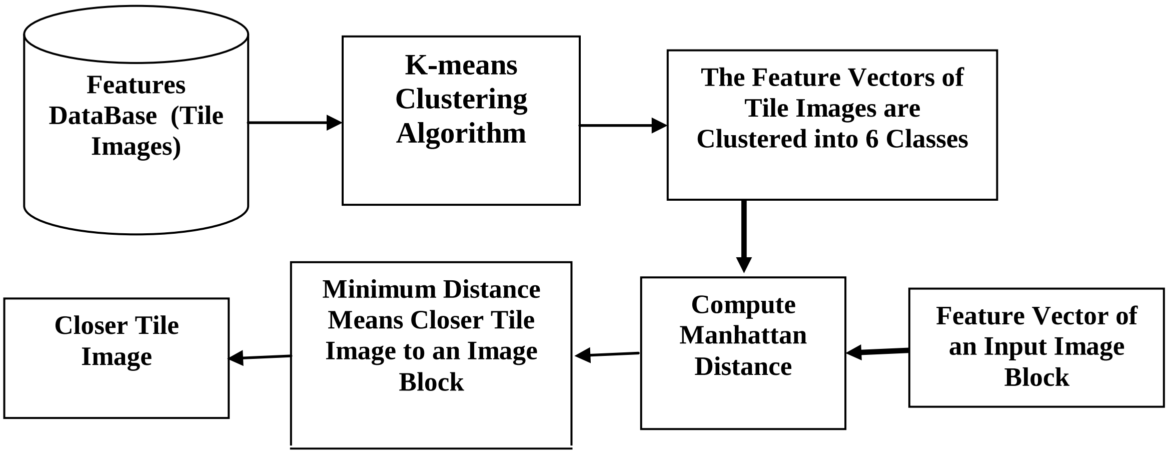

Features DataBase (Tile Images)
K-means Clustering Algorithm
The Feature Vectors of Tile Images are Clustered into 6 Classes
Compute Manhattan Distance
Feature Vector of an Input Image Block
Minimum Distance Means Closer Tile Image to an Image Block
Closer Tile Image

Fig. (2) Finding the closer tile image to an image block using K-means clustering algorithm and Manhattan distance

5.2.4 Back Propagation Neural Network with Manhattan Approach

BPNN consists of an input layer, one or more hidden layers, and an output layer. Connections among layers are typically formed by connecting each of the nodes in a given layer to all the neurons in the next layer [53]. In the training phase of BPNN, weight values are initialized with random values. The feature vectors of all tile images are set as input to BPNN, whereas the desired targets, which represent the classes of tile images, are set as output to BPNN. The obtained output value is compared with the desired output, followed by an error measurement and weight adjustment until the acceptable desired output for each feature vector is attained. To get the closest tile image to a given image block, an image block is classified to determine its proper class throughout the using phase of the trained BPNN. Then, the closest tile image from that proper class is derived using Manhattan distance Equation 15.

The architecture of the used BPNN consists of one input layer, one hidden layer, and one output layer. Nine neurons are found in the input layer, each one representing a feature of a feature vector; six neurons are located in the hidden layer; and one neuron is found in the output layer, which equals to one when the input feature vector belongs to class 1; two when the feature vector belongs to class 2; and so on, up to six, when the input feature vector belongs to class 6. Furthermore, hidden layer neurons are estimated using the hyperbolic tangent sigmoid transfer function, whereas the output layer neuron is estimated using the linear transfer function. The training algorithm used is gradient descent with momentum back propagation. The learning rate used is equal to 0.2 and the momentum constant used is equal to 0.8. Finally, the input and output data set of this study were normalized according to [54]. Figure (3) shows the proposed image retrieval algorithm based on BPNN with Manhattan distance.

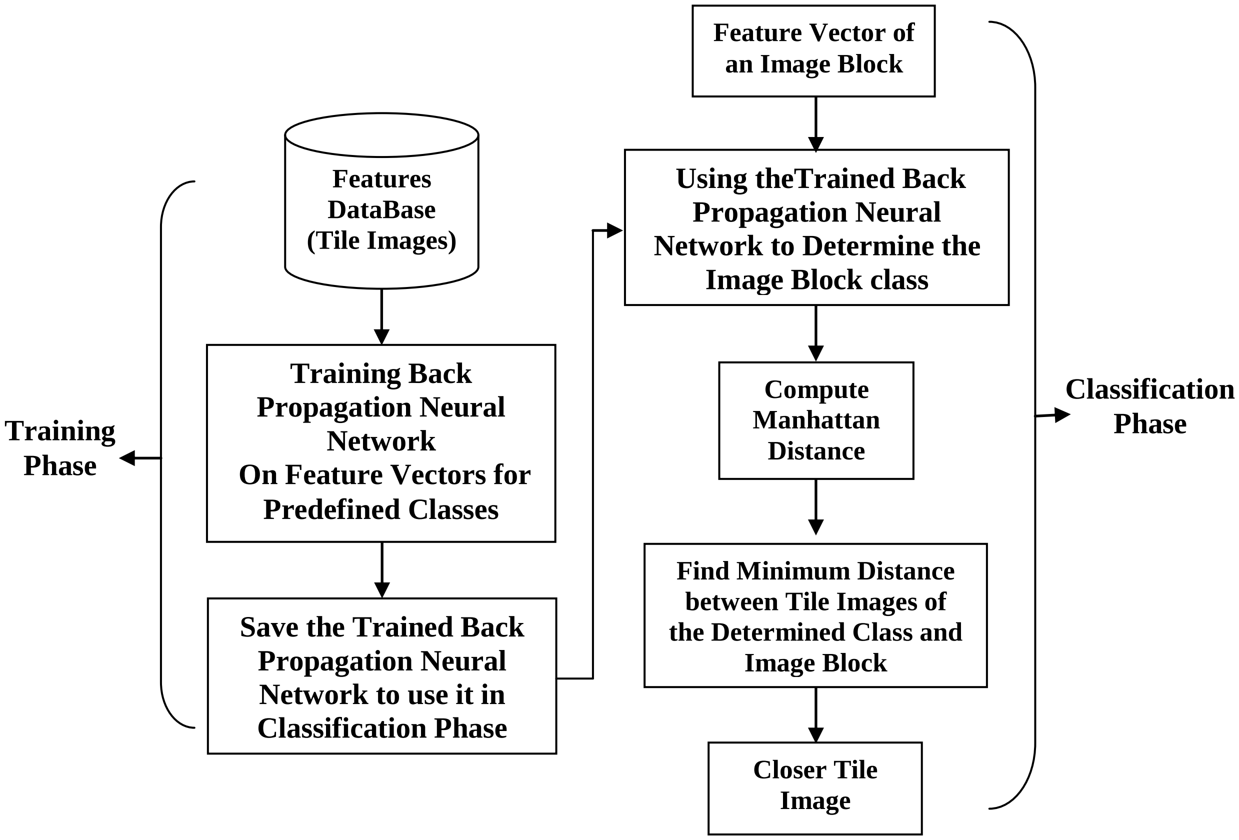

Feature Vector of an Image Block
Features DataBase (Tile Images)
Using theTrained Back Propagation Neural Network to Determine the Image Block class
Training Back Propagation Neural Network On Feature Vectors for Predefined Classes
Compute Manhattan Distance
Classification Phase
Training Phase
Find Minimum Distance between Tile Images of the Determined Class and Image Block
Save the Trained Back Propagation Neural Network to use it in Classification Phase
Closer Tile Image

Fig. (3) Finding the closer tile image to an image block using BPNN and Manhattan distance

5.2.5 Hybrid Fuzzy Logic with Elman Neural Network Approach

Two aspects are important in a fuzzy system: (1) generating the best rule set and (2) tuning the membership functions. These aspects should properly relate the independent and dependent variables [55].

The greatest advantage of using fuzzy logic is that it allows scientists to model non-linear, imprecise, and complex systems by transposing human experience, knowledge, and practice to inference (or fuzzy) rules that use linguistic (or fuzzy) variables [56][57]. A fuzzy set is a set in which the elements have degrees of membership. An element of a fuzzy set can be a full member (100% membership) or a partial member (between 0% and 100% membership). That is, the membership value assigned to an element is no longer restricted to only two values, but can be 0, 1, or any value in between. Mathematical function, which defines the degree of membership of an element in a fuzzy set, is called membership function [58]. A fuzzy system consists of a fuzzy variable described by its name tag, a set of fuzzy values, and the membership functions of these values. The membership functions assign a membership value to a given real value within a particular predefined range. Three basic operations, namely, “and,” “or,” and “not,” are used to combine different fuzzy variables to produce fuzzy logic expressions (rules) [59]. The fuzzy logic system involves three steps [56][57], that is, obtaining crisp results by:

1. Clearly specifying a set of fuzzy linguistic variables for the input or output system variables that describe it;
2. Defining a set of fuzzy inference rules between input and output fuzzy variables; and
3. Defining a membership function for each fuzzy variable.

Furthermore, a fuzzy logic system includes fuzzyfication and defuzzyfication processes. Fuzzification refers to the process of assigning numerical or crisp values to attributes with the help of membership functions. By contrast, defuzzyfication refers to the process of combining the degree of memberships of output variables to determine a numerical, crisp output value [60].

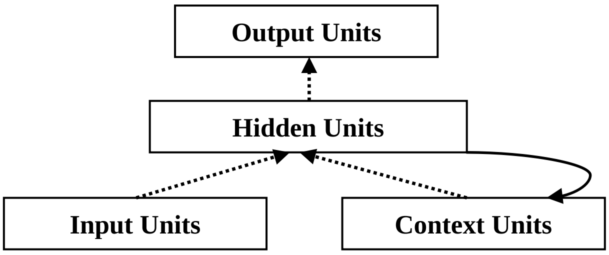


Fig.(4) A simple Elman recurrent neural network [45]

An Elman RNN [45] is a network which, in principle, is set up as a regular feed-forward network. As such, all neurons in one layer are connected with all neurons in the next layer with the exception of the context layer, which is a special case of hidden layer [61]. In general, the Elman RNN has three layer types: an input layer, a hidden layer, and an output layer. In addition, it also has special units called context units which save previous output values of hidden layer neurons [62]. The neurons in the context layer (context neurons) have a copy of the output of the hidden neurons. The output of each hidden neuron is copied into a specific neuron in the context layer. The value of the context neuron is used as an extra input signal for all neurons in the hidden layer one time step after [61]. The processing result in a previous time step can be used at the current time step [63]. Context unit values are then fed back fully connected, to hidden layer neurons. Thus, they serve as additional inputs to the network when output layer values are not fed back to the network [62]. Figure (4) shows a simple recurrent network in which activations are copied from a hidden layer to a context layer on a one-on-one basis, with a fixed weight of 1.0. The dotted lines represent trainable connections [45].

In this type of neural network, the weights from the hidden layer to the context layer are set to one and fixed because the values of the context neurons have to be copied exactly. Furthermore, initial output weights of the context neurons are equal to half of the output range of the other neurons in the network [61]. Through this research, we propose a hybrid system for photo-mosaic generation, which is a combination of fuzzy logic and Elman neural network. Fuzzy logic is used to determine a proper class for a given feature vector (image block), whereas Elman neural network determines the closest tile image to the image block, as shown in Figure(5).

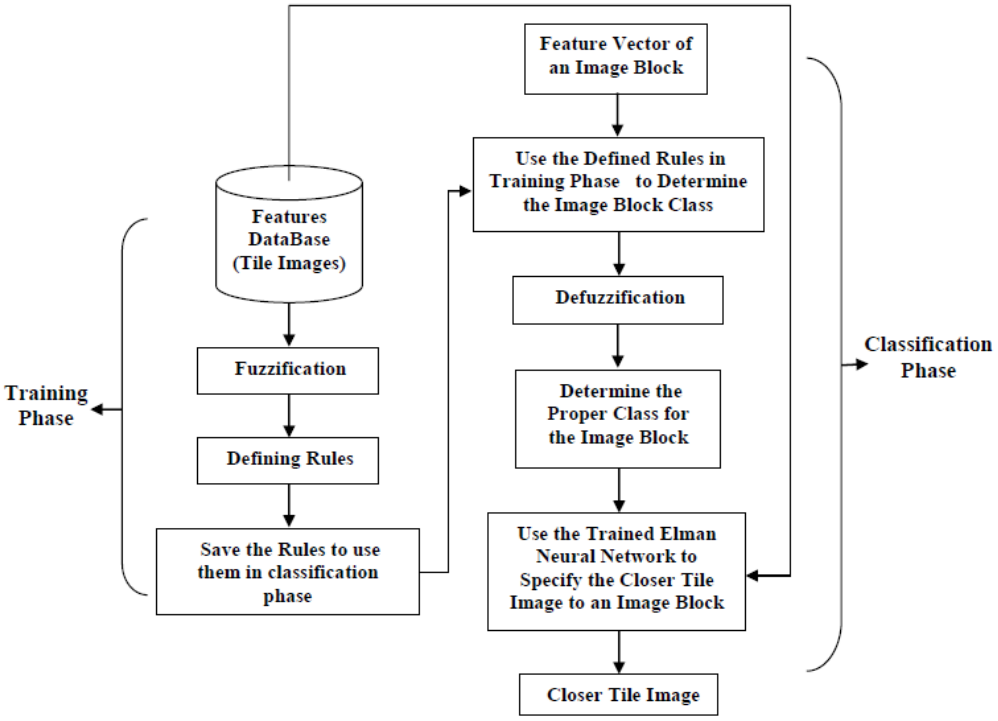

Feature Vector of an Image Block
Use the Defined Rules in Training Phase to Determine the Image Block Class
Features DataBase (Tile Images)
Defuzzification
Fuzzification
Classification Phase
Determine the Proper Class for the Image Block
Training Phase
Defining Rules
Save the Rules to use them in classification phase
Use the Trained Elman Neural Network to Specify the Closer Tile Image to an Image Block
Closer Tile Image

Fig. (5) Finding closer tile image to an image block using fuzzy logic and Elman NN

The database of feature vectors is classified into six classes, depending on their visual features (subjective features). For example, class one includes feature vectors of the airplane images. Four features (VARriance, KURtosis, COArseness, and PR2 percentage of edge pixels) are used to define the rules of the proposed photo-mosaic generator. The summary of the defined rules used in the fuzzy logic system is summarized in Table (1).

Table (1) Fuzzy Rule Definitions

Rule No.	**Rule Definition**
1	If (VAR is IL) and (KUR is IH) then (ClassNo is C1)
2	If (VAR is L) and (COA is H) and (PR2 is M) then (ClassNo is C1)
3	If (VAR is IH) and (KUR is L) and (PR2 is M) then (ClassNo is C2)
4	If (VAR is H) and (COA is M) then (ClassNo is C2)
5	If (VAR is IL) and (KUR is IL) and (COA is L) then (ClassNo is C3)
6	If (VAR is IH) and (KUR is IL) and (PR2 is L) then (ClassNo is C3)
7	If (VAR is VH) and (KUR is L) and (PR2 is L) then (ClassNo is C4)
8	If (VAR is H) and (COA is M) and (PR2 is L) then (ClassNo is C4)
9	If (VAR is VL) and (KUR is H) and (PR2 is M) then (ClassNo is C5)
10	If (VAR is IL) and (KUR is H) and (COA is M) then (ClassNo is C5)
11	If (VAR is L) and (KUR is H) and (PR2 is H) then (ClassNo is C6)
12	If (VAR is VL) and (COA is L) and (PR2 is H) then (ClassNo is C6)
M →Medium, L →Low, VL →Very Low, IL →Intermediate Low, H →High, VH →Very High and IH →Intermediate High	

The decision of selecting a closer tile image to an image block can be mapped into a three-layer Elman neural network. The input layer contains seven nodes which represent class number and values of six features (variance, kurtosis, coarseness, contrast, directionality, and percentage of edge pixels) of tile images that belong to a predetermined class. By contrast, the hidden layer contains six nodes, and the output layer contains a single node representing tile image number. The transfer function used is the hyperbolic tangent sigmoid. The Elman network was trained by gradient descent back-propagation with an adaptive learning rate. Furthermore, the learning rate used is equal to 0.8 and the momentum constant used is equal to 0.4. Finally, the input and output data set of Elman neural network were normalized according to [54].

6. Color Correction

After the tile images have been positioned in a particular arrangement based on color and texture features, the next task is to match the color (brightness, in case of gray-scale images) of the tile image to the color of the corresponding image block. The process of adjusting the colors of a tile image to match the colors of a corresponding image block is called color correction. Two algorithms are used for color correction. The first is inspired from [7], which is based on finding the average between the red, green, and blue colors of the image block and the red, green, and blue colors of the corresponding tile image. The second algorithm is inspired by histogram specification [64]. The algorithm of histogram equalization [65] is extended to histogram specification which can be used in the color-correction process. The color-correction process produces a new version of the tile image which has the same histogram as its corresponding image block. Given tile image T and image block B, if HT and HB are the equalizing transformations, then $\mathrm{H}_{\mathrm{B}}^{-1} \circ \mathrm{H}_{\mathrm{T}}(\mathrm{T})$ will be a new version of T with the same histogram as B [66].

7. Results

We propose three approaches for producing photo mosaics: (1) *k*-means clustering with Manhattan distance, (2) BPNN with Manhattan distance, and (3) hybrid fuzzy logic with Elman neural network.

Three input images (Lena, Monaliza, and Mandrill) and three different scales of tile images: 16 × 16, 32 × 32, and 64 × 64 pixels are used to test the proposed approaches for generating photo mosaics. Figure (6) shows the zoomed-in view of an example of a mosaic image produced via hybrid fuzzy logic and Elman neural network.

The algorithms of the three approaches are implemented using MATLAB 2009a; and all experiments are carried out on an Intel(R) Core

TM(2) Duo CPU, Memory (RAM) 3.00 GB, 32-bit Operating System, Windows Vista Home Edition, Service Pack 2.

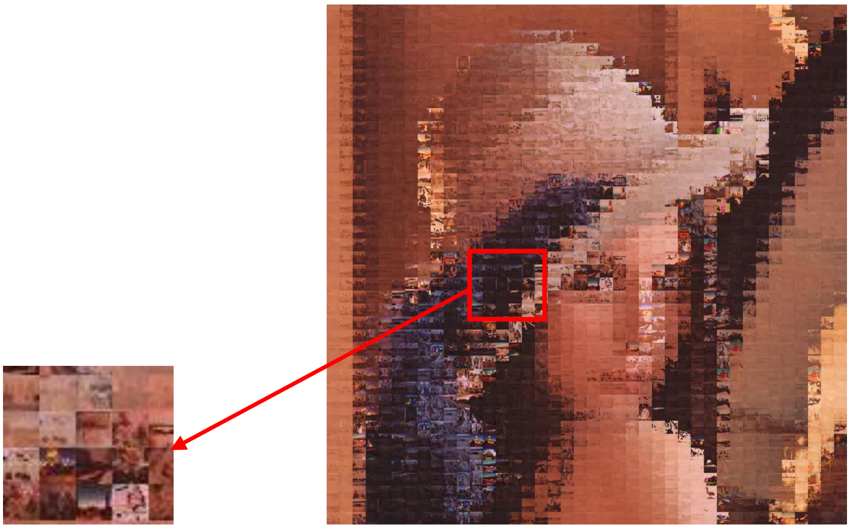
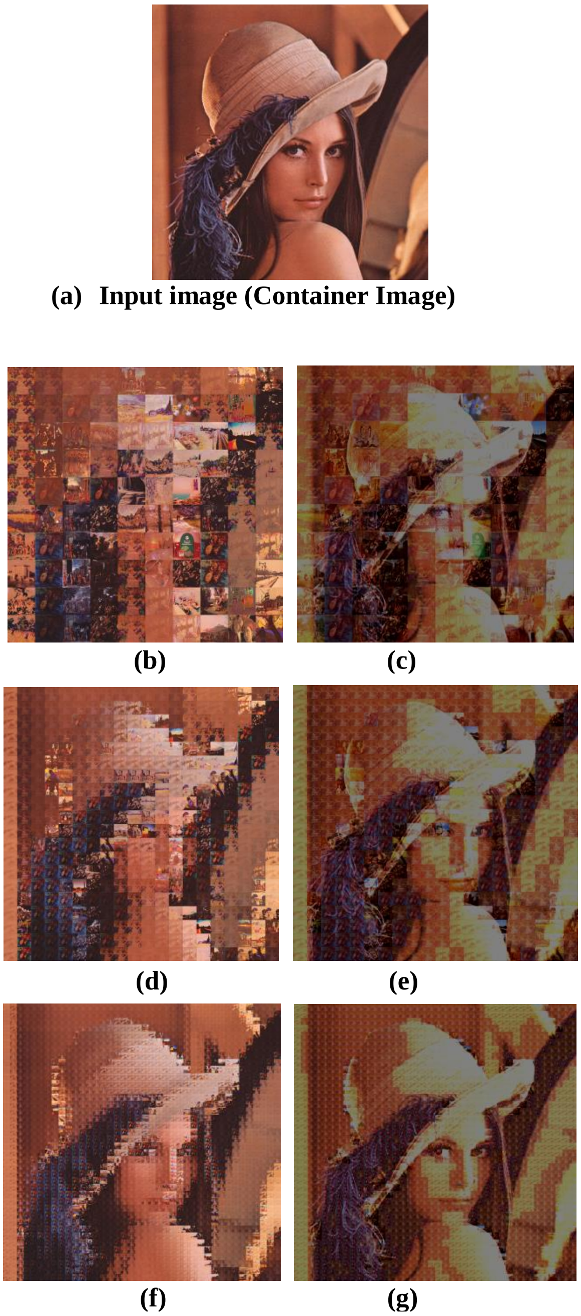


Figure (6) Zooming view of produced photo-mosaics using hybrid fuzzy logic & Elman NN

The obtained photo mosaics of the Lena image based on the *k*-means clustering with Manhattan distance approach using the three scales of the tile images are shown in Figures (7b, c, d, e, f, and g). Figure (7a) is the input image (container image). Figures (7b) and (7c) are obtained using 64 × 64-scale tile images, and color correction based on mean and histogram specification, respectively. Figures (7d) and (7e) are obtained using 32 × 32-scale tile images, and color correction based on mean and histogram specification, respectively. Finally, Figures (7f) and (7g) are obtained using 16 × 16-scale tile images, and color correction based on mean and histogram specification, respectively.



(a) Input image (Container Image)

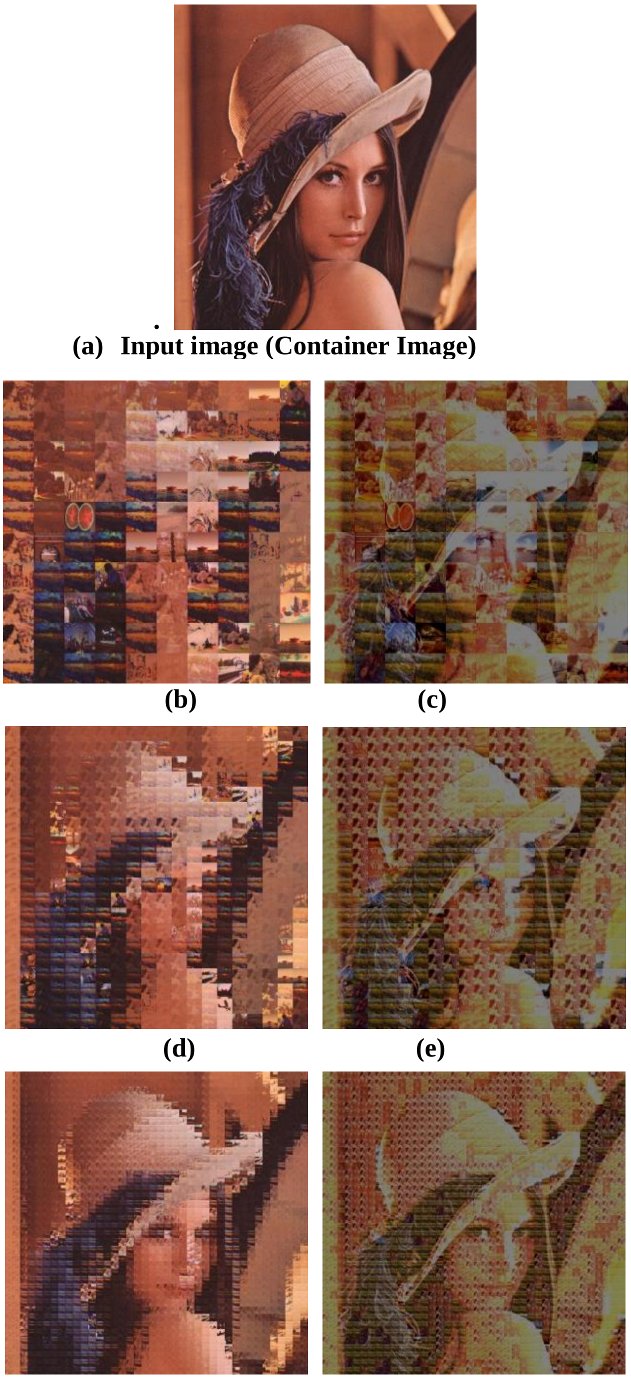
(b) (c)

(d) (e)

(f) (g)

Figure (7) The obtained photo-mosaics of Lena Image based on K-Means Clustering & Manhattan distance approach using the three scales of tile images

Figures (8 b, c, d, e, f, and g) show the generated photo mosaics of the Lena image based on the BPNN with Manhattan distance approach using the three scales of the tile images. The description for each figure is the same as that in the *k*-means clustering with Manhattan distance approach.



(a) Input image (Container Image)

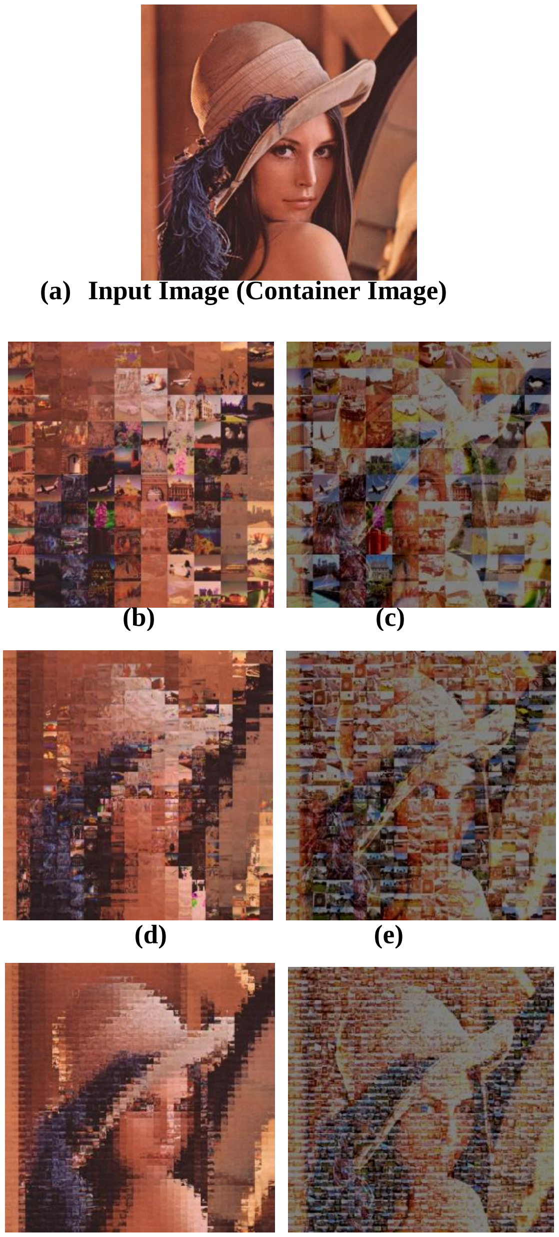
(b) (c)

(d) (e)

(f) (g)

Figure (8) The generated photo-mosaics of Lena image based on BPNN & Manhattan distance approach using the three scales of tile images

Finally, the produced photo mosaics of the Lena image based on the hybrid fuzzy logic with Elman neural network approach using the three scales of the tile images are shown in Figures (9b, c, d, e, f, and g). The description for each figure is the same as that in the *k*-means clustering with Manhattan distance approach.



(a) Input Image (Container Image)

(b) (c)

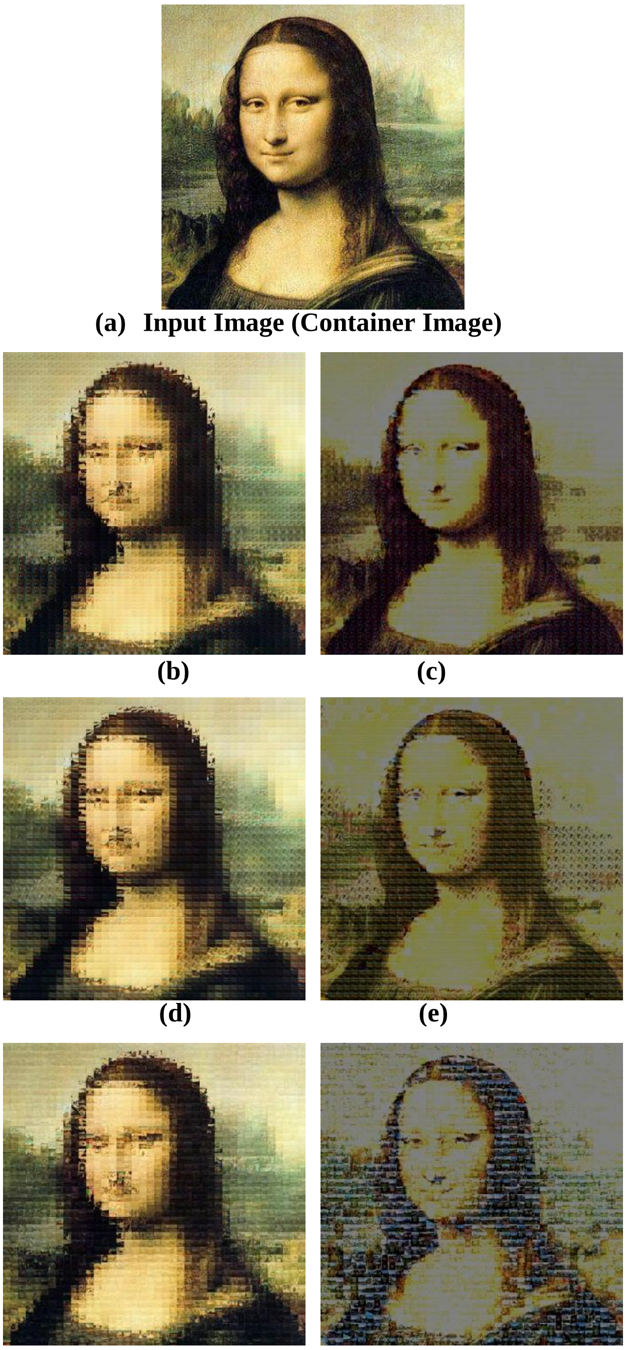
(d) (e)

(f) (g)

Figure (9) The generated photo-mosaics of Lena image based on Hybrid Fuzzy Logic and Elman NN approach using the three scales of tile images

The 16 × 16 scale of the tile images produced the best photo mosaics; therefore, we used this scale to test the three proposed approaches using the other two testing images (Monaliza and Mandrill).

Figures (10b, c, d, e, f, and g) show the produced photo mosaics of the Monaliza image based on *k*-means clustering with Manhattan distance, BPNN with Manhattan distance, and hybrid fuzzy logic with Elman neural network using 16 × 16-scale tile images, respectively. Photo mosaics shown in Figures (10c, e, and g) are produced via color correction based on mean, whereas those shown in Figures (10b, d, and f) are generated through color correction based on histogram specification.



(a) Input Image (Container Image)

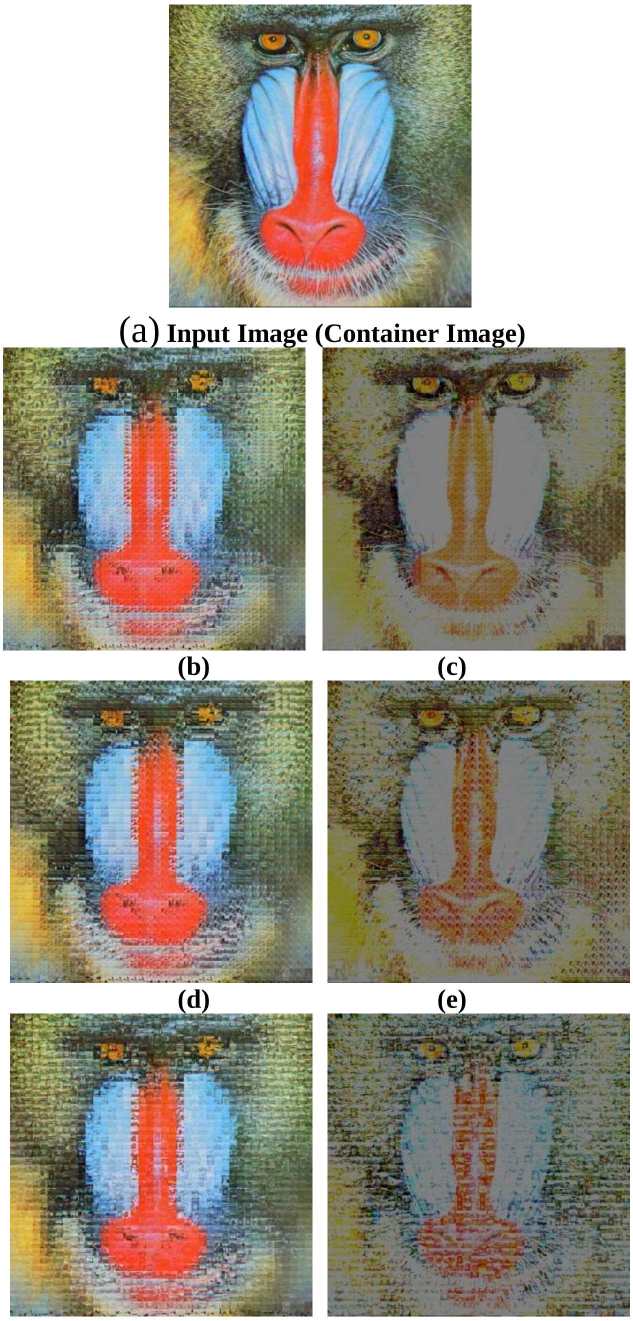
(b) (c)

(d) (e)

(f) (g)

Figure (10) The produced photo-mosaics of Monaliza image based on the three proposed approaches using 16 X 16 scales of tile images

The obtained photo mosaics of the Mandrill image based on *k*-means clustering with Manhattan distance, BPNN with Manhattan distance, and hybrid fuzzy logic with Elman neural network using 16 × 16-scale tile images are shown in Figures (11b, c, d, e, f, and g), respectively. The description for each figure is the same as those in Figures (10b, c, d, e, f, and g), respectively.



(a) **Input Image (Container Image)**

(b) **(c)**

(d) **(e)**

(f) **(g)**

Figure (11) The produced photo-mosaics of Mandrill image based on the three proposed approaches using 16 X 16 scales of tile images

Color correction based on histogram specification produced photo mosaics with colors more similar to the input images compared with photo mosaics obtained using mean-based color correction. Therefore, the histograms of the Lena image and the obtained Lena photo mosaics based on the three approaches using 16 × 16-scale tile images and color correction based on histogram specification are shown in Figure (12).

Figure (13) shows the histograms of the Monaliza image and the produced Monaliza photo mosaics based on the three approaches using 16 × 16-scale tile images and color correction based on histogram specification.

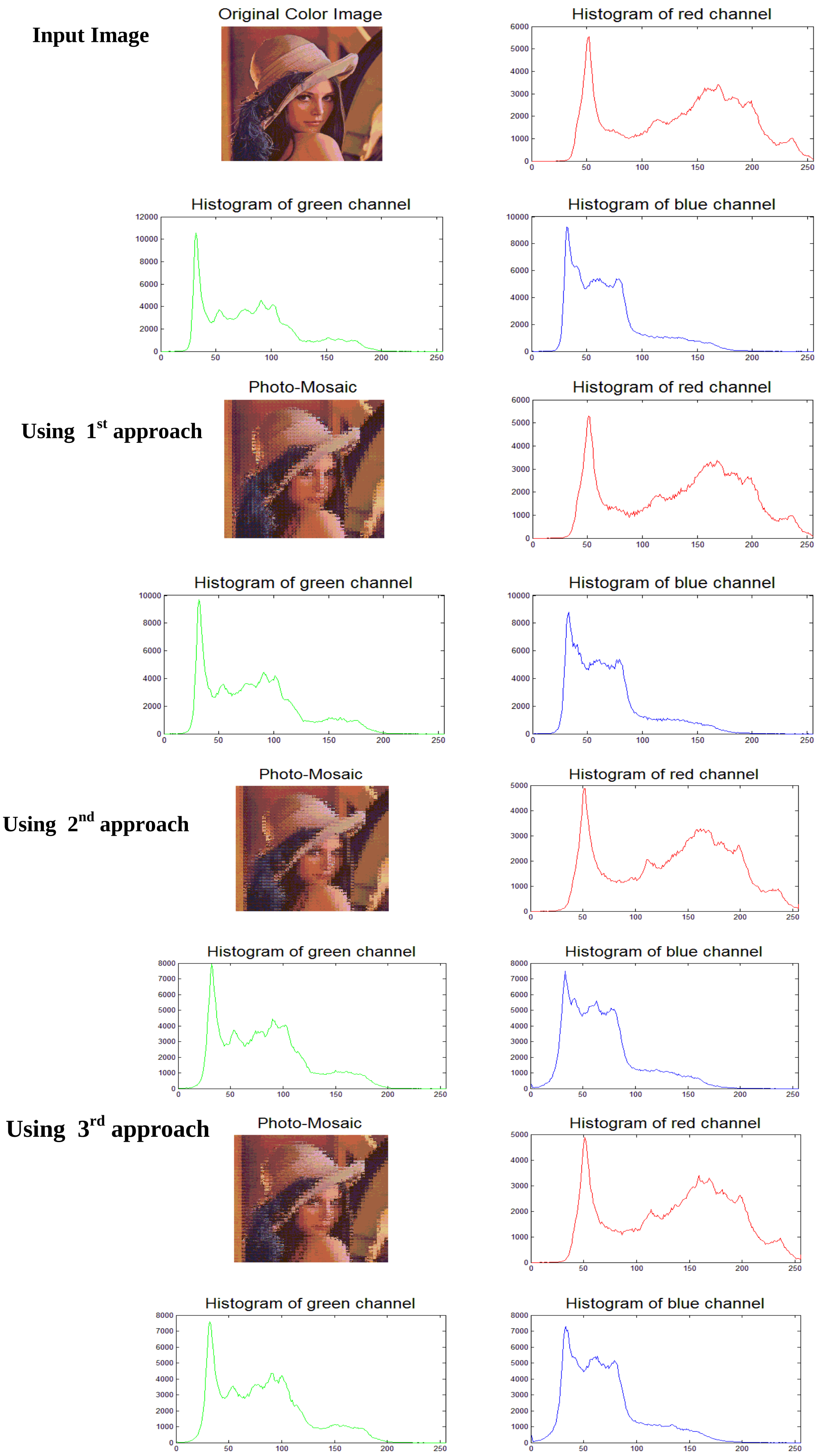


Figure (12) The histograms Lena image and the obtained Lena photo-mosaics

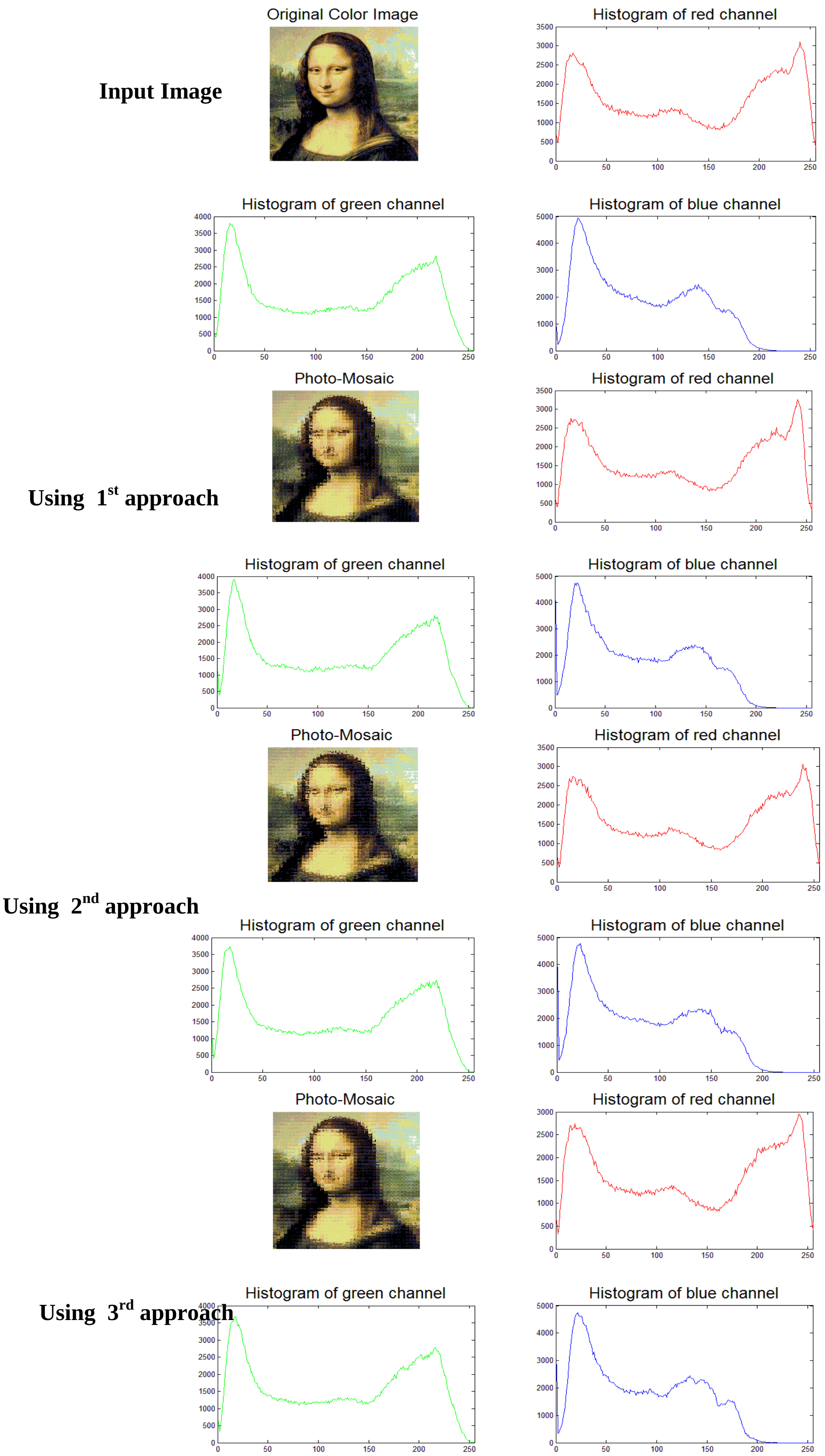


Figure (13) The histograms Monaliza image and the obtained Monaliza photo-mosaics

Finally, the histograms of the Mandrill image and the generated Mandrill photo mosaics based on the three approaches using 16 × 16-scale tile images and color correction based on histogram specification are shown in Figure (14).

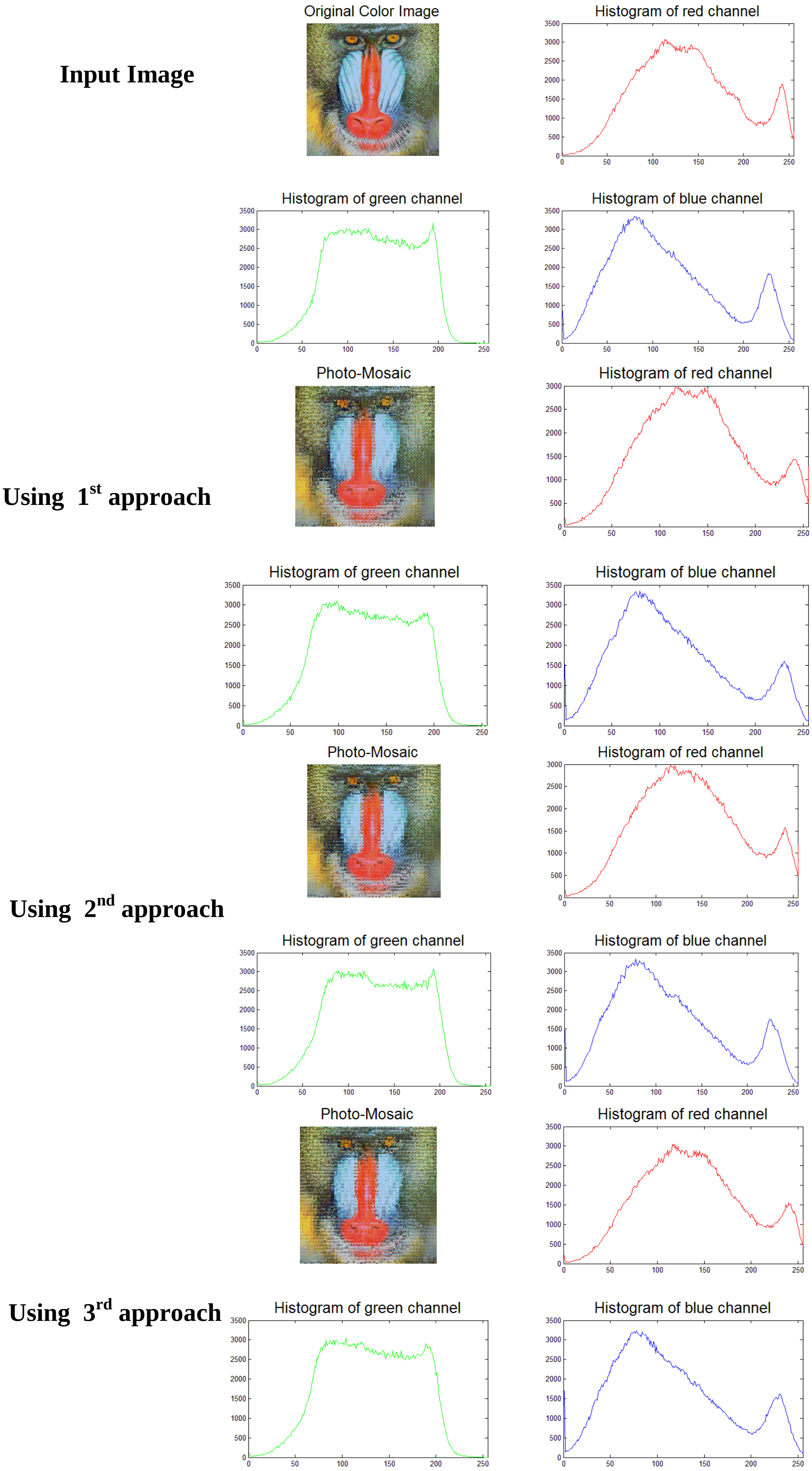


Figure (14) The histograms Mandrill image and the obtained Mandrill photo-mosaics

Average time results shown in Table (2) present the average total time to produce the three photo mosaics based on the three approaches using 16 × 16-scale tile images.

Table (2) Average total time of produced photo-mosaics using 16 X 16 scale tile images

Color Correction	1st approach	2nd approach	3rd approach
Mean	20.66 sec	83.33 sec	10.0 sec
Histogram Specification	53.0 sec	113.66 sec	42.33 sec

The average correlations between the container images and the corresponding photo mosaics for the three approaches using 16 × 16-scale tile images are illustrated in Table (3).

Table (3) Average total correlation between the origin images and produced photo-mosaics using 16 X 16 scale tile images

Color Correction	1st approach	2nd approach	3rd approach
Mean	0.81	0.80	0.82
Histogram Specification	0.84	0.83	0.86

The histograms shown in Figures (12), (13), and (14) provide visual representations of the correlation rates between the origin images and the obtained photo mosaics. In other words, the values of the average total correlation illustrated in Table (3) are comparable to the histograms shown in Figures (12), (13), and (14).

Figure (15) shows the average total time and total correlation of the three approaches using the three test images, 16 × 16-scale tile images, and mean-based color correction. Figure (16) presents the same details using histogram specification-based color correction.

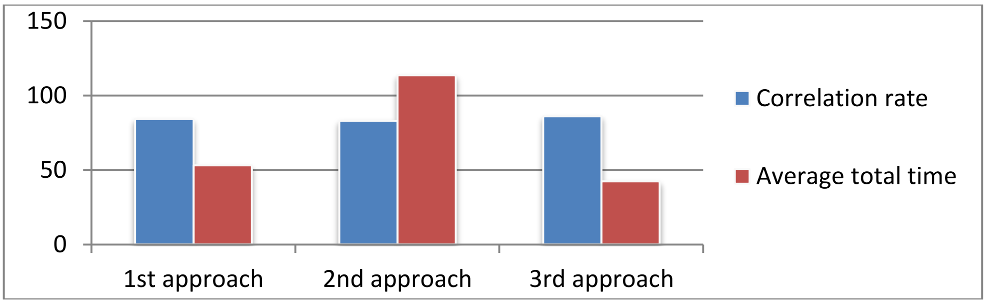


Figure (15) the averages of total time and total correlation of the three approaches using the three testing images, 16X16 scale tile images and mean based color correction.

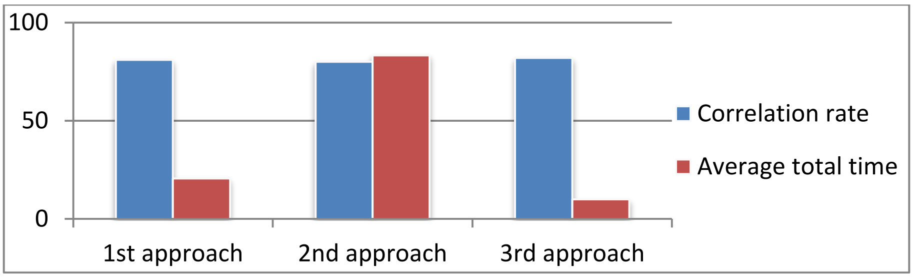


Figure (16) the averages of total time and total correlation of the three approaches using the three testing images, 16X16 scale tile images and histogram specification based color correction.

8. Discussion

Experimental results show that the hybrid fuzzy logic with Elman neural network approach exhibits the best performance evaluation over *k*-means clustering with Manhattan distance and BPNN with Manhattan distance approaches. The average time required to produce a photo mosaic of 16 × 16-scale tile images is 10.0 seconds and 42.33 seconds using mean-based color correction and histogram specification-based color correction, respectively. Moreover, the average total correlation between the origin images and the produced photo-mosaics of 16 × 16-scale tile images is 0.82 and 0.86 using mean-based color correction and histogram specification-based color correction, respectively. Furthermore, the histograms illustrated in Figures (12, 13 and 14) show the visual correlation between the origin image and the produced photo mosaics of 16 × 16-scale tile images.

The photo-mosaic generator is more commercial in nature, therefore only a few publications about it are available. A total time required among different photo mosaic generations are summarized in Table (4). The total mean time of the fast photo-mosaic generator proposed in [16] is 19.058 seconds for an origin image measuring 800 × 600 pixels without a color-correction step. In comparison, the total mean time of the artificial-mosaic generator proposed in [11] is 14.091 seconds for an origin image measuring 800 × 600 pixels without a color-correction step. The total time of the puzzle image-mosaic generator proposed in [5] is 267.131 seconds for an origin image measuring 600 × 600 pixels.

Table (4) Required time among different photo-mosaic generations

Approach	Image size	Color Correction	Required time
[16]	800 X 600	Without	19.058 seconds
[11]	800 X 600	Without	14.091 seconds
[5]	600 X 600	Without	267.131 seconds
Proposed 3rd approach	640 X 640	Mean based	10.0 seconds
Proposed 3rd approach	640 X 640	Histogram specification based	42.33 seconds

The resulting high correlation of the hybrid fuzzy logic with Elman neural network approach using 16 × 16-scale tile images and histogram specification-based color correction is comparable to those of existing commercial photo-mosaic generators.

9. Conclusions

In this research, three different approaches to produce photo-mosaics are presented: (1) *k*-means clustering with Manhattan distance, (2) BPNN with Manhattan distance, and (3) hybrid fuzzy logic with Elman neural network. These approaches use three types of features to find the corresponding tile image of an image block. The first group of features includes statistical features extracted from a 64-gray-level quantized histogram. These features are variance, mean, skewness, kurtosis, and energy. The second group of features includes Tamura features such as coarseness, contrast, and directionality. The last feature is edge rate, which is computed as the percentage of edge pixels within an image detected using a Canny filter.

Timing results (Table (2)) show that the third is the fastest among the three approaches, needing only 10.0 s and 42.33 s to produce a photo mosaic using mean- and histogram specification-based color correction, respectively. Correlation rate results (Table (3)) demonstrates the soundness of the third approach, which achieves good visual impact. The third approach provides the highest correlation rates between the input image and the produced photo mosaic, which are 0.82 and 0.86 when using mean- and histogram specification-based color correction, respectively.

10. Future works

This research suggests a number of directions for further studies. The photo-mosaic framework can be expanded to three-dimensional (3-D) mosaic, wherein the input and the tile images can be 3-D to fill out the surface of the input. The framework can also be expanded to produce other kinds of photo mosaics, such as ancient mosaics. Moreover, extracting shape-based features to select tile images that best match an image block can also be explored.

References

[1] Jain, A. K. (1989). Fundamentals of digital image processing., Englewood Cliffs, NJ, Prentice hall.

[2] Foley, J. D. (1996). Computer graphics: principles and practice. Addison-Wesley Professional.

[3] Parent, R., (2002), Computer animation: algorithms and techniques, Academic Press,USA

[4] Bordwell, D., Thompson, K., & Ashton, J. (1986). Film art: an introduction . McGraw-Hill.

[5] Di Blasi, G., Gallo, G., & Petralia, M. (2005, September). Puzzle image mosaic. In Proc. IASTED/VIIP2005.

[6] Di Blasi, G., Gallo, G., & Petralia, M. (2006). Smart ideas for photomosaic rendering. In Proceedings of Eurographics Italian Chapter Conference.
[7] Finkelstein, A., & Range, M. (1998). Image mosaics. Electronic Publishing, Artistic Imaging, and Digital Typography, pp. 11-22.
[8] Battiato, S., Di Blasi, G., Farinella, G. M., & Gallo, G. (2007, June). Digital Mosaic Frameworks-An Overview. In Computer Graphics Forum (Vol. 26, No. 4, pp. 794-812). Blackwell Publishing Ltd.
[9] Lai, I. J., & Tsai, W. H. (2011). Secret-Fragment-Visible Mosaic Image–A New Computer Art and Its Application to Information Hiding. Information Forensics and Security, IEEE Transactions on, 6(3), pp. 936-945.
[10]Kim, J., & Pellacini, F. (2002). Jigsaw image mosaics. ACM Transactions on Graphics, 21(3),pp. 657-664.
[11] Di Blasi, G., & Gallo, G. (2005). Artificial mosaics., The Visual Computer,21(6),pp.373-383.
[12] Silvers, R. S. (2000). U.S. Patent No. 6,137,498. Washington, DC: U.S. Patent and Trademark Office
[13] Lai, Y. K, Shi-Min Hu and Ralph R. Martin (2006), Surface Mosaics, The Visual Compute,Vol. 22, Issue 9-11, pp 604-611.
[14] Kim, J., Pellacini, F.: Jigsaw image mosaics. ACM Transactionson Graphics 21(3), 657–664 (2002)
[15] Tauheed, F. (2004, December). Photo collage and image similarity quantification. In Engineering, Sciences and Technology, Student Conference On (pp. 127-131). IEEE.
[16] Di Blasi, G., & Petralia, M. (2005, January). Fast photomosaic. In poster proceedings of ACM/WSCG2005.
[17] Mikamo, M., Slomp, M., Yanase, S., Raytchev, B., Tamaki, T., & Kaneda, K. (2010, November). Maximizing Image Utilization in Photomosaics. InNetworking and Computing (ICNC), 2010 First International Conference on (pp. 275-278). IEEE.
[18] Mat Sah, S., Ciesielski, V., & D’Souza, D. (2010). Refinement techniques for animated evolutionary photomosaics using limited tile collections. Applications of Evolutionary Computation, 281-290.
[19] Lee, S. H., Kim, B. S., Moon, Y. S., & Kim, J. Photo Mosaic Smartphone Application in Client-Server Based Large-Scale Image Databases.
[20] Miller, J., & Mould, D. (2012, June). Accurate and discernible photocollages. In Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging (pp. 115-124). Eurographics Association.

[21] Salehian, H., Zamani, F., & Jamzad, M. (2012). Fast Content Based Color Image Retrieval System Based on Texture Analysis of Edge Map. Advanced Materials Research, 341, 168-172.

[22] Kekre, H. B. , Sonawane K. (2012). "Statistical Moments Extracted from Eight Bins Formed by CG Partitioning of Histogram Modified using Linear Equations",IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 2, pp.421-430

[23] Tamura, H., Mori, S., & Yamawaki, T. (1978). Textural features corresponding to visual perception. Systems, Man and Cybernetics, IEEE Transactions on,8(6), 460-473.

[24] Canny, J. (1986). A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, (6), 679-698.

[25] Ojala, T., & Pietikäinen, M. (2001). Texture classification. CVonline: The Evolving, Distributed, Non-Proprietary, On-Line Compendium of Computer Vision.

[26] Kodituwakku, S. R., & Selvarajah, S. (2004). Comparison of color features for image retrieval. Indian Journal of Computer Science and Engineering, 1(3), 207-211.

[27] Srinivasan, G. N., & Shobha, G. (2008, December). Statistical texture analysis. In Proceedings of world academy of science, engineering and technology (Vol. 36, pp. 1264-1269).

[28] Tuceryan, M., & Jain, A. K. (1998). Texture analysis. Chapter 2.1 in Handbook of pattern recognition and computer vision, (2nd Edition), by C. H. Chen, L. F. Pau, P. S. P. Wang (eds.), pp. 207-248, World Scientific Publishing Co.

[29] Otsu, N. (1979). A threshold selection method from gray-level histograms. IEE Transcation on Biomedical Engineering Vpl. 9, pp. 63-66.

[30] Materka, A., & Strzelecki, M. (1998). Texture analysis methods-a review.Technical University of Lodz, Institute of Electronics, COST B11 report, Brussels, 9-11.

[31] Shahbahrami, A., Borodin, D., & Juurlink, B. (2008, November). Comparison between color and texture features for image retrieval. In Proceedings of the 19th Annual Workshop on Circuits, Systems and Signal Processing, ProRisc.

[32] Deselaers, T. (2003). Features for image retrieval. Rheinisch-Westfalische Technische Hochschule, Technical Report, Aachen.

[33] Liu, Y., Zhang, D., Lu, G., & Ma, W. Y. (2007). A survey of content-based image retrieval with high-level semantics. Pattern Recognition, 40(1), 262-282.

[34] Howarth, P., & Rüger, S. (2004). Evaluation of texture features for content-based image retrieval. Image and Video Retrieval, 2134-2135.

[35] R. Keys, (1981). "Cubic convolution interpolation for digital image processing". IEEE Transactions on Signal Processing, Acoustics, Speech, and Signal Processing 29 (6): 1153–1160

[36] Malik, F., & Baharudin, B. B. Feature Analysis of Quantized Histogram Color Features for Content-Based Image Retrieval Based on Laplacian Filter, (2012), International Conference on System Engineering and Modeling (ICSEM 2012) , 7th - 8th April 2012,Kuala Lumpur, Malaysia, IPCSIT vol. 34.

[37] Clausi, D. A. (2001). Comparison and fusion of co-occurrence, Gabor and MRF texture features for classification of SAR sea-ice imagery. Atmosphere-Ocean,39(3), 183-194.

[38] L.G. Shapiro and G.C. Stockman. (2001),Computer Vision. Prentice Hall.

[39] Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24(7), 881-892.

[40] Liu, H., & Yu, X. (2009). Application Research of k-means Clustering Algorithm in Image Retrieval System. In Proceedings of the Second Symposium International Computer Science and Computational Technology (ISCSCT'09)(pp. 274-277).

[41] Vadivel, A. K. M. S. S. A., Majumdar, A. K., & Sural, S. (2003). Performance comparison of distance metrics in content-based image retrieval applications. In Proc. of Internat. Conf. on Information Technology, Bhubaneswar, India (pp. 159-164).

[42] Kong, W., Li, W. J., & Guo, M. (2012, August). Manhattan hashing for large-scale image retrieval. In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval (pp. 45-54). ACM.

[43] Hecht-Nielsen, R. (1989, June). Theory of the backpropagation neural network. In Neural Networks, 1989. IJCNN., International Joint Conference on (pp. 593-605). IEEE.

[44] Godara, S., & Gupta, R. (2013). Neural Networks for Iris Recognition: Comparisons between LVQ and Cascade Forward Back Propagation Neural network Models, Architectures and Algorithm. Neural Networks, 3(1).

[45] Elman, J. L. (1990). Finding structure in time. Cognitive science, 14(2), 179-211.

[46]Kainijo, K. I., & Tanigawa, T. (1990). Stock price pattern recognition-a recurrent neural network approach. In Proc. of Int. Conf. Neural Network

[47]Wang, H., & Gao, Y. (2011). Elman's Recurrent Neural Network Applied to Forecasting the Quality of Water Diversion in the Water Source of Lake Taihu.Energy Procedia, 11, 2139-2147

[48]Chamorro-Martinez, J., Galán-Perales, E., Sánchez, D., & Soto-Hidalgo, J. (2006). Modelling coarseness in texture images by means of fuzzy sets. InKnowledge-Based Intelligent Information and Engineering Systems (pp. 355-362). Springer Berlin/Heidelberg.

[49]Ahmed, H. A., El Gayar, N., & Onsi, H. (2008). A New Approach in Content-Based Image Retrieval Using Fuzzy Logic.

[50] Pham, T., Bro, P., & Troncoso, J. L. (2010, December). Fuzzy analysis of color images of fruits. In Information Science and Engineering (ICISE), 2010 2nd International Conference on (pp. 3714-3717). IEEE.

[51] Abdel-Mottaleb, M., Krishnamachari, S., & Mankovich, N. (1998, June). Performance evaluation of clustering algorithms for scalable image retrieval. InProc. IEEE Workshop on Empirical Evaluation Techniques in Computer Vision.

[52] Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001, June). Constrained k-means clustering with background knowledge. In MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE- (pp. 577-584).

[53] Jyothi, B. V., & Shanker, U. (2010). Neural Network approach for image retrieval based on preference elicitation. International Journal, on Computer Science and Engineering vol. 2, Issue: 4 Pages: 934-941

[54] Aksoy, S., & Haralick, R. M. (2001). Feature normalization and likelihood-based similarity measures for image retrieval. Pattern Recognition Letters,22(5), 563-582.

[55]Shailendrakumar, M. M., Sachin, R. G., & Dattatraya, S. B. (2011). On Scale Invariance Texture Image Retrieval using Fuzzy Logic and Wavelet Co-occurrence based Features. International Journal of Computer Applications, 18(3), pp. 10-17.

[56] Balanică, V., Dumitrache, I., Caramihai, M., Rae, W., & Herbst, C. (2011). Evaluation Of Breast Cancer Risk By Using Fuzzy Logic. U.P.B. Sci. Bull., Series C, Vol. 73, Iss. 1.

[57] Caramihai, M., Severin, I., Blidaru, A., Balan, H., & Saptefrati, C. (2010, August). Evaluation of breast cancer risk by using fuzzy logic. In Proceedings of the 10th WSEAS international conference on applied informatics and communications, and 3rd WSEAS international conference on Biomedical electronics and biomedical informatics (pp. 37-42). World Scientific and Engineering Academy and Society (WSEAS).

[58] Nedeljkovic, I. (2004). Image classification based on fuzzy logic. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Commission VI, Vol. 34 (pp. 1−6) Part XXX.

[59] Pena-Reyes, C. A., & Sipper, M. (1999). A fuzzy-genetic approach to breast cancer diagnosis. Artificial intelligence in medicine, 17(2), 131-155.

[60] Lehner, K., & Hartmann, D. (2007). Using knowledge based fuzzy logic components in the design of underground engineering structures. Proc., 1st EURO: TUN (Computational Methods in Tunneling).

[61] Übeyli, E. D. (2009). Combining recurrent neural networks with eigenvector methods for classification of ECG beats. Digital Signal Processing, 19(2), pp. 320-329.

[62]Karjala, T. W., Himmelblau, D. M., & Miikkulainen, R. (1992, June). Data rectification using recurrent (Elman) neural networks. In Neural Networks, 1992. IJCNN., International Joint Conference on (Vol. 2, pp. 901-906). IEEE.

[63] Wang, H., & Gao, Y. (2011). Elman's Recurrent Neural Network Applied to Forecasting the Quality of Water Diversion in the Water Source of Lake Taihu.Energy Procedia, 11, pp. 2139-2147.

[64]Coltuc, D., Bolon, P., & Chassery, J. M. (2006). Exact histogram specification. Image Processing, IEEE Transactions on, 15(5), 1143-1152.

[65]Cheng, H. D., & Shi, X. J. (2004). A simple and effective histogram equalization approach to image enhancement. Digital signal processing, 14(2), 158-170.

[66] Pichon, E., Niethammer, M., & Sapiro, G. (2003, September). Color histogram equalization through mesh deformation. In Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on (Vol. 2, pp. II-117). IEEE.