Int. J. Nonlinear Anal. Appl. 14 (2023) 1, 1331–1337
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.27446.3602

A segmentation-based image zooming algorithm using artificial neural networks

Mohammad M. AlyanNezhadi[a,*], Ashkan Fakhri[a], Farzan Afshari[b]

[a] *Faculty of Science, University of Science and Technology of Mazandaran, P.O. Box 4851878195, Behshahr, Iran*

[b] *Faculty of Physics, Semnan University, Semnan, Iran*

(Communicated by Javad Vahidi)

Abstract

This paper aims to present a useful method for magnifying images, for which it is necessary to group the pixels and define the borders. In the proposed method, images are first partitioned using suitable segmentation algorithms and then artificial neural networks (ANNs) are applied to magnify each segment individually. In the ANNs applied, training is performed using, as input, a down-sampled form of the same image to be magnified. This type of training results in a high quality zoom in each segment since the pixels in an individual segment have very close features. Evaluation results on several images verifies the higher efficiency of the proposed method than other recently developed image zooming methods.

Keywords: Artificial Neural Network, Machine Learning, Multi-Layer Perceptron, Image processing, Image Zooming; Image Segmentation
2020 MSC: Primary 68T01; Secondary 68T05

1 Introduction

In many applications, it is necessary to use zoom functions to observe a specific area of an image in more detail. However, the zoom functions enlarge the corresponding area of the image using some simple algorithms, such as linear interpolation. The zoomed image quality is usually quite poor and blurry and hence, proposing new efficient zooming algorithms is a very important and challenging research area.

Generally, a zooming algorithm takes, as input, an RGB image and provides, as output, an image of double size preserving as much as possible the features of the original image. A large number of conventional zooming algorithms use some kinds of linear interpolation. Replication, nearest neighbor, bilinear and bicubic algorithms are the most usual such algorithms that are routinely applied in commercial image processing software.

In conventional methods an interpolation function is applied indiscriminately to the whole image. No matter how complex the chosen function is, these algorithms have a main drawback that they introduce jaggedness in previously smooth images and undesirably reduces contrast (sharp edges).

On the other hand, nonlinear interpolation or adaptive methods, aim at avoiding these problems by analyzing the local structure of the input image and using different interpolation functions over different areas of the image.

*alyan.nezhadi@mazust.ac.ir
Email addresses: a.fakhri@mazust.ac.ir (Ashkan Fakhri), farzanafshari@outlook.com (Farzan Afshari)

Received: January 2022 *Accepted:* May 2022

Ahmed et al [1] presented an adaptive algorithm for image resampling based on segmenting the image dynamically into homogeneous areas and preserving edge locations. Based on the location of the pixel to be computed (within a homogenous area, on its edge or an isolated feature), three options can be considered to choose among: interpolation, extrapolation or pixel replication.

Edge-directed interpolation (EDI) algorithms (see [2], [9] and [5] for example) were presented that aim to preserve edges in the image after zooming and prevent introducing staircase artifacts. The idea behind EDI methods is to make sure that the image is not smooth perpendicular to edges and it is smooth parallel to edges. These methods use statistical quantities, and in particular, variances, to produce information about edges as high local variance quantities means large changes in value i.e. an edge. These quantities are then used to adapt the interpolation. Performance of some EDI methods are evaluated and compared in [10];

Another important adaptive method is the locally adaptive zooming (LAZ) [4]. The basic idea of LAZ algorithm is to use a gradient-controlled, weighted interpolation.

Furthermore, a number of image zooming methods have been developed using partial differential equations (PDEs) [8] to preserve edges and minimize blurring and staircase effects in the zoomed images.

Finally, artificial neural networks (ANNs) use machine learning for resampling more detailed images such as photographs and complex artwork. [6] [7]

In the present paper, we propose an adaptive image zooming method based on segmentation of the image and using an ANN over each individual segment. The applied ANN uses machine learning techniques with a down-sampled form of the original image as input.

The rest of the paper is organized as follows. In Section 2, we discuss on the importance of segmentation in image zooming and we introduce some segmentation methods. Our proposed adaptive zooming method is presented in Section 3. Evaluation of our method and comparisons with other methods are reported in Section 4. Finally, we conclude the paper in Section 5.

2 Segmentation

One of the key steps in the analysis of digital images is the definition of its borders. This step is known as segmentation and this is a processing technique employed to analyze and to group pixels according to their features and attributes.

There are plenty of segmentation methods effective in specific kinds of images. The following are some well-known segmentation methods (all available in MATLAB):

(i) Thresholding

These techniques perform thresholding on a grayscale image to create a binary image. To produce a binary image from an RGB color image, one may first convert it to a grayscale image.

(ii) Clustering

These techniques partition an image using a specific clustering algorithm. K-means clustering–based segmentation algorithms partition an image into K number of clusters. Mixture models are another type of clustering techniques. A mixture model is a probabilistic model for identifying subpopulations within an overall population.

(iii) Graph-Based Segmentation

Graph-based segmentation techniques enable us to segment an image into foreground and background regions.

(iv) Region Growing

Region growing methods are simple region-based (also classified as a pixel-based) image segmentation methods. Active-countour is a popular algorithm which iteratively adds the neighbors of initial seed points to the region.

(v) Deep Learning for Image Segmentation

Deep learning techniques within Convolutional neural networks (CNNs) are very good for classifying all kinds of images, but their main drawback is that they need a lot of data to be well trained. Medical imaging, autonomous driving, and satellite image analysis are some applications of this segmentation method.

3 The proposed method

The image zooming method of this paper, includes four main steps as follows:

Step 1. Segmentation

First, the original image T is called to perform a segmentation on it yielding a predefined number of image segments $T^1, T^2, ..., T^k$. The purpose of this stage is to increase the quality of network training. Indeed, instead of applying the Multi-Layer Perceptron (MLP) network only once to the whole image, we apply it several times, once for each segment of T.

In our zooming method, we apply the Gaussian mixture model (GMM) in which the Expectation Maximization (EM) algorithm is used to estimate the parameters of the distribution. GMMs assume that there are a certain number of Gaussian distributions, each representing a cluster. Indeed, a GMM tends to group the pixels belonging to a single distribution.

Step 2. Down-sampling

At the second stage, a down-sampling (by factor 0.5) is performed on the image T to obtain a reduced image T_D. Pixels of T_D are then partitioned into segments T_D^i with respect to the segments of T (i.e. T^i, i=1,2, ...,k).

Step 3. Training

In order to train the ANN, all pixels of an individual segment T_D^i along with their neighboring pixels (in a 3×3 window) are used as input data and the corresponding 2×2 block of pixels in T^i are considered as output data in a separate MLP network. Then, this network is trained to magnify the related segment by a factor of 2.

The used MLPs have one hidden layer, nine neurons in the input layer and four neurons in the output layer. Each neuron in one layer connects with certain weights to every neuron in the subsequent layer. The value of a neuron in the hidden layer is calculated by executing an appropriate function (f^1) on the sum of the weighted combination of the neurons of the input layer and a bias. Then, The value of a neuron in the output layer is calculated by executing another appropriate function (f^2) on the sum of the weighted combination of the neurons of the hidden layer and some bias. Each layer's weights and biases are initialized in random. After each certain number of passes of the training data through the MLP, the weights are updated in the reverse direction by means of a Gradient Descent algorithm. Usually, f^1 is considered to be the following Log-sigmoid function:

$$f^1(x) = \frac{1}{1 + e^{-x}}$$

which projects R onto (0, 1). Table 1 summarizes the features of the MLP used in this paper.

Table 1: Basic features of the MLP used.

Subject	Value
Number of Neurons in Input Layer	9
Number of Neurons in Hidden Layer	10
Number of Neurons in Output Layer	4
Transfer Function of Hidden Layer	Log-sigmoid transfer function
Transfer Function of Output Layer	Linear transfer function
Training Method	Bayesian regularization backpropagation

Step 4. Up-sampling

The trained MLPs are then employed to magnify the segments of the original image T. As aforementioned, the training stage is performed once for each image (segment), while the trained network can be used as many times as desired, each time for magnifying the image by a factor of 2.

Figure 1 provides (by an example) an overview of how our proposed method works while Figure 2 shows its intermediate operations to magnify an individual segment.

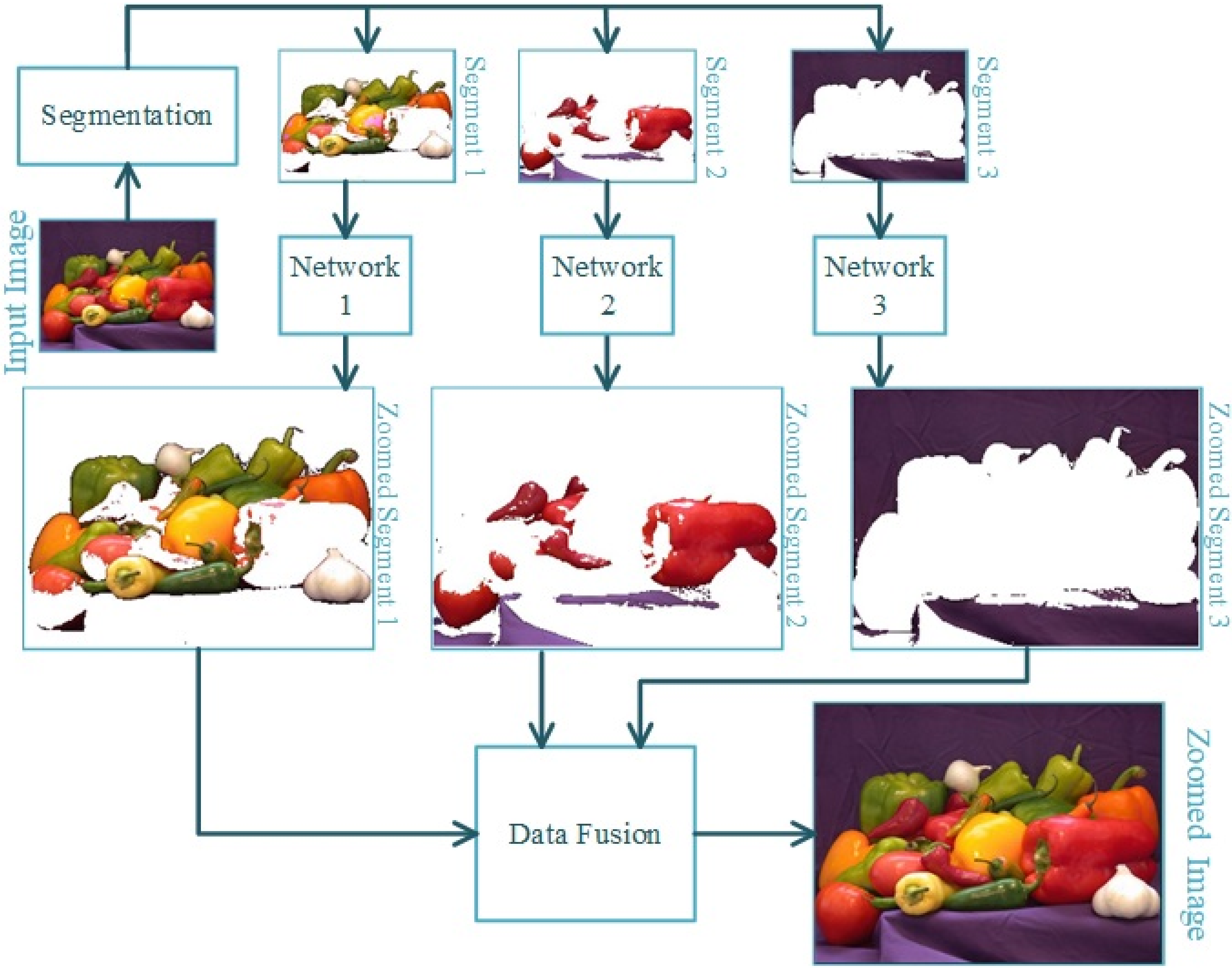

Figure 1: An overview of the proposed method

4 Evaluation

In this section, first we introduce some well-known criteria for image quality assessment and then we evaluate our zooming method with respect to these criteria.

4.1 Metrics

Two types of metrics can be applied for image quality assessment: subjective metric (or human judgement) and objective metrics (or quantitative criteria). In this paper, we use objective metrics to evaluate the performance of the proposed method. The objective methods used in our evaluation are briefly described below.

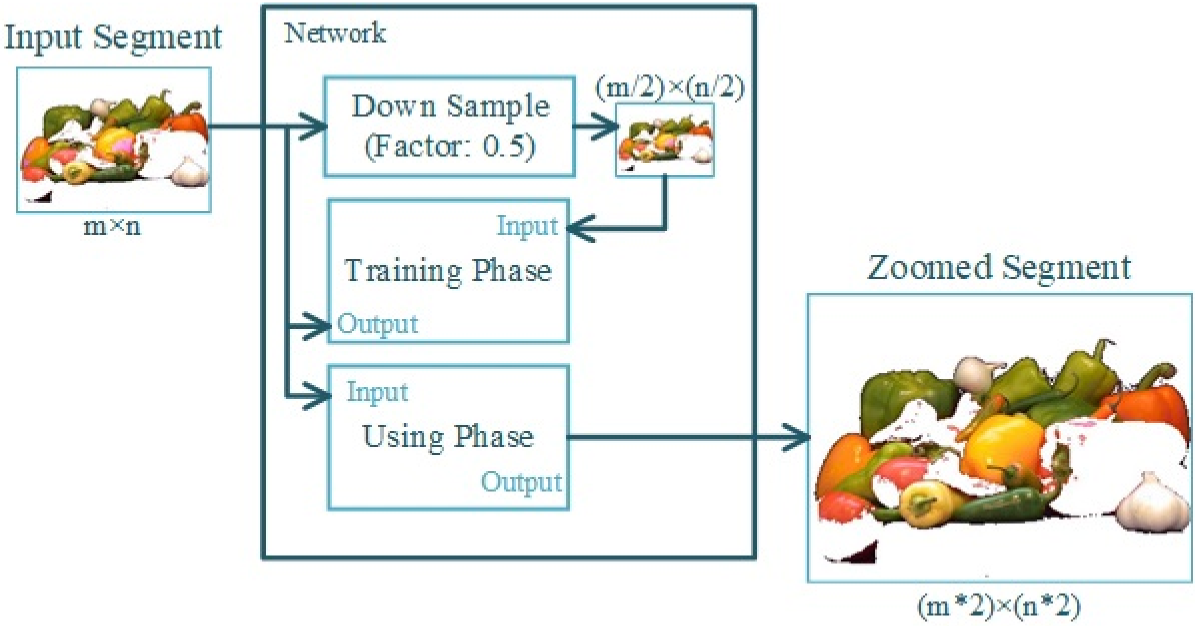

Figure 2: Magnifying an individual segment by MLP network

4.1.1 Mean Squared Error (MSE)

MSE is a well-known criterion to compare two images. For two images A and B, we sum the squares of the differences between every pixel of A and the corresponding pixel in B, and then divide the result by the total number of pixels:

$$MSE(I_{org}, I_{dis}) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij}^{o} - x_{ij}^{d})^2$$

where, I_{org} and I_{dis} denote the original image and the distorted image, respectively. Also, x^o and x^d represent the value of pixels in I_{org} and I_{dis}, respectively, and the two images are assumed to be of size m by n. A value closer to zero of MSE can be considered as a closer resemblance between the two images.

4.1.2 Peak Signal-to-Noise Ratio (PSNR)

PSNR of two images I_{org} and I_{dis} is defined by the following formula:

$$PSNR(I_{org}, I_{dis}) = 10log_{10}(\frac{255^2}{MSE(I_{org}, I_{dis})})$$

The value 255 in this formula is the maximum possible pixel value when the pixels are represented using 8 bits per sample. The higher the PSNR, the closer resemblance between the two images.

4.1.3 Figure of merit (FOM)

FOM is a metric that measures the quality of the output image with respect to its edges and is defined as follows:

$$FOM(I_{org}, I_{dis}) = \frac{1}{max(N_{org}, N_{dis})}\sum_{i=1}^{N_{dis}}\frac{1}{1 + d_i^2\lambda}$$

where, N_{dis} and N_{org} are the edge pixels of the distorted and the original images, respectively. d_i is the Euclidean distance between the i-th edge pixel in the distorted image and the nearest edge pixel in the original image. The constant λ is usually set to 1.9. The closer the FOM value is to 1, the greater the edge retention in the distorted image.

4.1.4 Structural Similarity (SSIM)

SSIM is a perception-based model that quantifies degradation of structural information of images. SSIM is formulated as below:

$$SSIM(I_{org}, I_{dis}) = \frac{(2\mu_{I_{org}}\mu_{I_{dis}} + c_1)(2\sigma_{I_{org},I_{dis}} + c_2)}{(\mu^2_{I_{org}} + \mu^2_{I_{dis}} + c_1)(\sigma^2_{I_{org}} + \sigma^2_{I_{dis}} + c_2)}$$

where, μ and σ^2 are the average and the variance of the values of pixels, respectively.

SSIM generates a value in the interval [0,1]. The higher this value, the more structural similarity between the original and distorted images.

4.2 Results

In order to evaluate the performance of the proposed method, we used a number of images to run the method on them. We considered these images as our 'references' to evaluate our method. First, we performed the segmentation phase on these images. We down-sampled the images (by factor 0.5) and identified the relevant segments in the down-sampled images. Then, we considered these down-sampled images as the 'original' images to be magnified. Hence, again, a down-sampling (by factor 0.5) was performed on the original images to produce doubly shrunk images with identified segments. Then, each individual segment of a doubly shrunk image was considered separately as the input for an MLP network while the corresponding segment in the original image was considered as its output, and the network was trained. Finally, the trained MLPs were applied to magnify their corresponding segments of the original image. The outputs of these networks were combined and the resultant was compared to the reference image using the metrics presented in Sub-section 4.1. Tables 2-4 summarize the comparison results. All methods were implemented in Matlab R2016a installed on a PC with 2.3 GHz Intel Core i3 CPU, 4GB RAM and 64-bit Windows 8 operating system.

Table 2: Comparison results w.r.t. PSNR

	Im1	Im2	Im3	Im4	Im5	Im6
Proposed Method	30.37	34.36	27.11	33.98	23.29	32.19
Ref. [7]	30.35	34.35	27.09	33.93	23.28	32.17
Ref. [3]	30.39	34.06	27.06	33.84	23.24	32.17
The nearest interpolation method	28.96	29.43	24.11	30.97	22.39	28.61
The bilinear interpolation method	29.27	30.53	24.44	31.90	22.26	29.99
LAZ [4]	28.25	28.63	23.28	30.02	21.72	27.61
AZ [2]	28.11	28.43	23.11	29.84	21.58	27.43
NPDE [8]	25.48	26.59	21.54	25.76	20.49	24.84

Table 3: Comparison results w.r.t. FOM

	Im1	Im2	Im3	Im4	Im5	Im6
Proposed Method	0.9766	0.9837	0.9829	0.9866	0.9771	0.9860
Ref. [7]	0.9765	0.9814	0.9819	0.9860	0.9771	0.9855
Ref. [3]	0.9750	0.9829	0.9826	0.9858	0.9770	0.9852
The nearest interpolation method	0.9622	0.9709	0.9722	0.9702	0.9596	0.9721
The bilinear interpolation method	0.9284	0.9377	0.9484	0.9401	0.9252	0.9425
LAZ [4]	0.9074	0.9135	0.9214	0.9112	0.9068	0.9121
AZ [2]	0.8960	0.9012	0.9144	0.9019	0.8946	0.9031
NPDE [8]	0.9130	0.8704	0.8707	0.9064	0.9222	0.9046

As can be seen from the tables 2-4, the method proposed in this paper outperforms the other methods in almost all cases (with respect to various metrics).

Table 4: Comparison results w.r.t. SSIM

	Im1	Im2	Im3	Im4	Im5	Im6
Proposed Method	0.9766	0.9917	0.8596	0.9889	0.8311	0.9812
Ref. [7]	0.9736	0.9916	0.8593	0.9888	0.8308	0.9810
Ref. [3]	0.9736	0.9910	0.8594	0.9886	0.8294	0.9812
The nearest interpolation method	0.9650	0.9769	0.8157	0.9806	0.7909	0.9667
The bilinear interpolation method	0.9650	0.9802	0.8064	0.9828	0.7625	0.9726
LAZ [4]	0.9594	0.9712	0.7876	0.9778	0.7472	0.9611
AZ [2]	0.9576	0.9693	0.7788	0.9767	0.7346	0.9592
NPDE [8]	0.9413	0.9627	0.7501	0.9646	0.7633	0.9426

5 Concluding Remarks

In this paper, an image zooming method was presented based on segmentation of the image and using separate MLP neural networks on each individual segment. For training the ANNs applied, nothing is required but a down-sampled form of the segments of the same image to be magnified. This type of training results in a high quality zoom in each segment since the pixels in an individual segment have very close features. The evaluation results show the superior performance of the proposed method in almost all test instances. Determining the best number of segments for partitioning the image to be magnified can be the subject a future study. Furthermore, improving techniques for magnifying the boundaries of the segments can be considered for increasing the quality of the proposed method.

References

[1] M. Ahmed, M. Darwish and M.S. Bedair, *Adaptive resampling algorithm for image zooming*, Proc. SPIE 2666, Image and Video Process. IV, 1996, https://doi.org/10.1117/12.234736.

[2] J. Allebach and P.W. Wong, *Edge-directed interpolation*, Image Process. Proc. IEEE **16** (1996), 707—710.

[3] M.M. AlyanNezhadi and F. Afshari, *Applying zoom on ultrafast ultrasound localization microscope image of the cerebral small vessel: Towards a better diagnosis*, 6th Conf. Electric. Comput. Engin. Technol., Tafresh, Iran, March, 2022, pp. 9–10.

[4] S. Battiato, G. Gallo and F. Stanco, *A locally adaptive zooming algorithm for digital images*, Image Vision Comput. **20** (2002), no 11, 805–812.

[5] Z. Dengwen, S. Xiaoliu and W. Dong, *Image zooming using directional cubic convolution interpolation*, IET Image Process. **6** (2012) no. 6, 627–634.

[6] Y. Douzi, M. Benabdellah, A. Azizi, T. Hajji and M. Jaara, *Zoom and restoring of digital images with artificial neural networks*, Comput. Sci. Engin. **5** (2015), no. 1, 14–24.

[7] H. Hassanpour, N. Nowrozian, M.M. AlyanNezhadi and N. Samadiani, *Image zooming using a multi-layer neural network*, Comput. J. **61** (2018), no. 11, 1737—1748.

[8] H. Hassanpour, *Image zooming using non-linear partial differential equation*, Int. J. Engin. **27** (2014), no. 1, 15–28.

[9] X. Li and M.T. Orchard, *New edge-directed interpolation*, IEEE Trans. Image Process. **10** (2001), no. 10, 1521–1527.

[10] S. Yu, R. Li, R. Zhang, M. An, S. Wu and Y. Xie, *Performance evaluation of edge-directed interpolation methods for noise-free images*, Proc. Fifth Int. Conf. Internet Multimedia Comput. Service, August, 2013, pp. 268—272.