Grayscale Images and RGB Video: Compression by Morphological Neural Network

Osvaldo de Souza[1], Paulo César Cortez[1], and Francisco A.T.F. da Silva[2]

[1] Federal University of Ceará, DETI, Fortaleza, Brazil
osvaldo@ufc.br, cortez@lesc.ufc.br
[2] National Institute For Space Research, ROEN, Eusébio, Brazil
tavares@roen.inpe.br

Abstract. This paper investigates image and RGB video compression by a supervised morphological neural network. This network was originally designed to compress grayscale image and was then extended to RGB video. It supports two kinds of thresholds: a pixel-component threshold and pixel-error counting threshold. The activation function is based on an adaptive morphological neuron, which produces suitable compression rates even when working with three color channels simultaneously. Both intra-frame and inter-frame compression approaches are implemented. The PSNR level indicates that the compressed video is compliant with the desired quality levels. Our results are compared to those obtained with commonly used image and video compression methods. Network application results are presented for grayscale images and RGB video with a 352 × 288 pixel size.

Keywords: Supervised Morphological Neural Network, RGB Video Compression, and Image Compression.

1 Introduction

The loss of data is common in a variety of image and video compression techniques, and such losses generally occur in parts of the information (redundancy data) that are not noticed by human eyes. Numerous compression algorithms utilize common techniques such as *"color space sampling"* and *"redundancy reduction"* [1]. The color space sampling technique is used when it is necessary to reduce the amount of data needed for the representation (coding) of an image. In the redundancy reduction technique, compression can be obtained by eliminating the redundancies that appear in a specific frame (intra-frame) or in a sequence of frames (inter-frame) in a video stream. Several studies have investigated the use of artificial neural networks (ANN) in image and video compression [1]. Some researchers [2] investigated image compression and reconstruction using a radial basis function (RBF) network, while others [3] proposed a technique called a "point process" that used a combination of motion estimation, compression, and temporal frame sub-sampling with a random neural network (RNN). In [4], the authors discussed various ANN architectures for

N. Mana, F. Schwenker, and E. Trentin (Eds.): ANNPR 2012, LNAI 7477, pp. 213–224, 2012.
© Springer-Verlag Berlin Heidelberg 2012

image compression and presented the results for a back-propagation network (BPN), hierarchical back-propagation network (HBPN), and adaptive back-propagation network (ABPN). In [5], they used a self-organizing map (SOM) network to reduce the number of pixels in each frame of a video sequence. After this modification, each frame was stored using a Hopfield neural network as a form of video codification. In [6], they used the growing neural gas (GNG) learning method, another approach based on a SOM network, in an incremental training algorithm. In [7], the authors presented the details of an approach in which a neural network is used to determine the best ratio for discrete cosine transform (DCT) compression. Although there have been many works related to image and video compression, the use of supervised morphological neural networks (SMNNs) in this context has not been extensively investigated thus far. Therefore, in this paper, we investigate the extension of an SMNN, which was originally designed to compress grayscale images, by applying it to the compression of RGB video.

We organize the remaining sections of this paper as follows. We first provide a brief review of the morphological operators involved in the design of the adaptive morphological neuron. Second, we introduce the SMNN for grayscale image compression and then extend its application to RGB video compression. Third, we present the image and video compression results.

2 Brief Review of Morphological Operators

The morphological operators presented in this section were defined in [8] and briefly in [9], while the researchers in [10] proposed a morphological approach for template matching.

Definition 1. Let E be a non-void set in $\mathbb{z}^2$ and l be an integer number between 0 and n. $K_n^E \; in \; K_l^E$, denoted by $\delta_l \; \varepsilon_l \; \varepsilon_l^a \; \delta_l^a$, are operators defined as dilation, erosion, anti-dilation, and anti-erosion, respectively. Formal definitions for these operators are given in [8] and [10].

Definition 2. Let a window W be a non-void subset of $\mathbb{Z}^2$. An individual element of W is denoted by w, according to:

$$w \in W \mid W \subset \mathbb{Z}^2. \tag{1}$$

Definition 3. Let $W \subset \mathbb{Z}^2$ be a window and D and E be two non-void subsets of $\mathbb{Z}^2$, such that $D = E \oplus W$, and let l be an integer number between 0 and m. The symbol $\oplus$ refers to Minkowski addition. We denote by ε_l^i and δ_l^{ai} the operators from K_m^D to K_1^E defined in [10].

Definition 4. Let $f \in K_n^W$ and c_1 and c_2 $(c_1 < c_2)$ be two integer constants. We define the following functions from W to K_n:

$$(f_w^-)(x) = max\{0, min(n, f_w(x) + c_1)\}, \tag{2}$$

$$(f_w^+)(x) = max\{0, min(n, f_w(x) + c_2)\}, \tag{3}$$

where $x \in W$. The values of c_1 and c_2 are calculated according to de following equation.

$$d\mu = \mu(g) - \mu(f) \qquad c_1 = d\mu - \frac{F}{2} \qquad c_2 = d\mu + \frac{F}{2} \tag{4}$$

where the length of F is the interval $[c_1, c_2]$, centered at $d\mu$.

Definition 5. Let ε_l^i and δ_l^{ai} be the operators from K_m^D to K_1^E given by definition 1, and let f_w^- and f_w^+ be the functions defined by (2) and (3), respectively. λ^i is an operator from K_m^D to K_1^E defined by:

$$\lambda^i = \varepsilon^i_{f_w^-(w_i)} \wedge \delta^{ai}_{f_w^+(w_i)}, \tag{5}$$

where W is a window; $m = \#W; i = 1, \ldots, m$.

Definition 6. The pattern matching operator from K_m^D to K_m^E is:

$$\phi = \sum_{i=1,\ldots,n} \lambda^i \tag{6}$$

The operator ϕ represents the intersections between erosions and anti-dilations with the tolerances introduced by equations (2) and (3), which are controlled by the value of F. Therefore, such operations start to behave as morphological operators with gray level tolerance. Observe that the operations in equation (6) result in adaptive pattern matching.

Definition 7. Let $t \in K_n$ be a threshold. The operator $\Psi_t^\bullet$ from K_n^E to K_1, which localizes a concentration of gray levels above or equal to t, according to [11] is defined as:

$$\Psi_t^\bullet \circ \phi = \begin{cases} 1, & if\ \exists\ x\ \in E, \qquad f(x) \geq t \\ 0, & otherwise \end{cases} \tag{7}$$

This threshold operator is a morphological filter, which is useful for adaptive pattern detection, and it is a key component in the activation function of the morphological neuron used in this work, as discussed later. The equations and definitions presented in this section are first applicable to gray scale images. Thus, it is important to note that because the color components in RGB schema can be represented in a range of values between 0 to 255, all of the definitions and proofs available in [8-10] and [12] that were developed for gray scale images are suitable for processing color images, since we consider only one component of the color at a time. This strategy is adopted in this work, and we refer to the values between 0 to m of the color component, as the "color variation of the component" (CVC), where $0 < m < 255$.

3 SMNN for Image Compression

The activation potential and activation function of SMNN for image compression are based on equations (6) and (7), and they are defined according to:

$$v_k = \phi, \tag{8}$$

and

$$y_k = \Psi_t^{\bullet} \circ \phi. \tag{9}$$

During the supervised training, the weights first decay in order to accelerate the weight adjustment process, according to equation (10):

$$W_{k_n} = W_{k_{n-1}} \left(1 - \left(\varepsilon * \Delta_{m_n}\right)\right). \tag{10}$$

Where $0 < \varepsilon < 1$, W_{k_n}refers to the weight of neuron k in iteration n, and Δ_m is defined according to

$$\Delta_m = 1 - \left(\left(\sum_{i=1}^{MN} A\right) \Big/ \Delta max_m\right). \tag{11}$$

Then the weights are adjusted according to

$$W_{k_n} = W_{k_{n-1}} + \beta \times \left(D_k - W_{k_{n-1}}\right), \tag{12}$$

where D_k is the desired value, and β Is an array, defined by

$$\beta = \mu\rho. \tag{13}$$

Note that μ is the learning constant and ρ is an array defined by the following equations

$$A = \left\{\varepsilon^{i}_{f_{\bar{W}}(w_i)} \wedge \delta^{ai}_{f^{+}_{W}(w_i)}\right\}, \tag{14}$$

where A is an array with dimensions K_1^W.

$$V_e = A^C, \tag{15}$$

where A^C is the complement of A,

$$\Delta max_m = M * N, \tag{16}$$

where M and N are the dimensions of A^C, and finally we have

$$\rho = \Delta_m V_e. \tag{17}$$

The activation function for the morphological neuron in the auxiliary layer is:

$$\delta^*_{D_k}(y_k) = \begin{cases} f_{D_k}, & if\ y_k = 1 \\ 0_D, & otherwise \end{cases} \tag{18}$$

Note that equation (18) is a morphological dilation. Figure 1 presents the architecture for this SMNN; observe that the network is composed of an input layer, an output layer, and a hidden layer with its auxiliary layer. The input layer receives the patterns to be learned; in this case, “patterns” refers to the data of the image to be compressed.

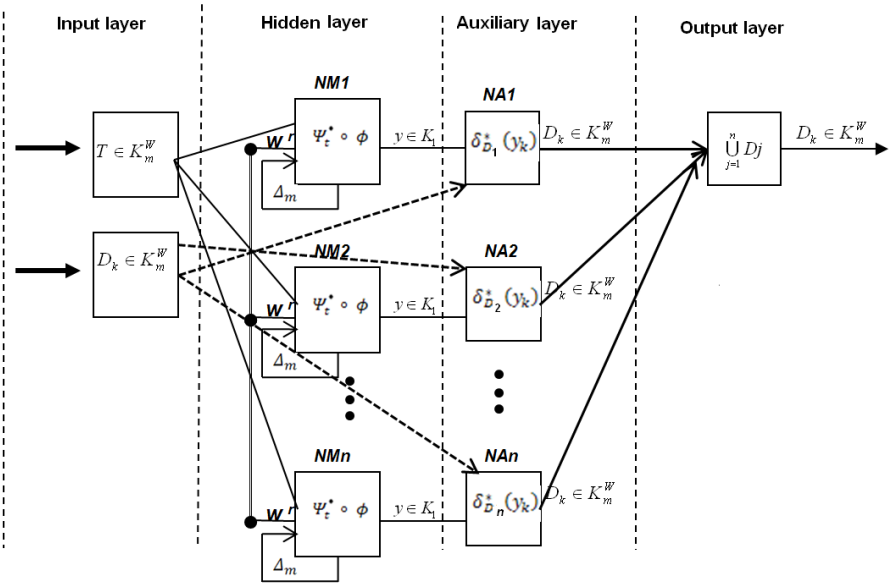

Fig. 1. Architecture of a SMNN

In this architecture, the T patterns presented to the input layer must belong to the K_m^W domain; the D_k value refers to the desired value, which is mandatory for error correction in the learning phase. The network's output is limited to the K_m^W domain. K_m^W refers to a sub-image of dimension W with a positive grayscale level between [0, m]. $y \in K_1$ refers to the output of a morphological neuron of the second layer.

According to definition 4 and equations (2–4) SMNN allows the definition of a pixel-component threshold, in fact, the tolerance interval F. This interval defines the tolerance of the SMNN to deal with gray level variations between the neurons' weights and a pattern under processing by the network. In addition, SMNN also allows the definition of a pixel-error counting threshold. This threshold is responsible for restricting the neuron's activation. In the following sections, we extend a SMNN in order to make it capable of compressing images and RGB video.

3.1 Grayscale Image Compression

The image to be compressed is fragmented into a set of windows. Each element of this set is processed by all of the morphological neurons (MNs). The Winner neuron produces an output with value 1 (high), while all of the others produce outputs with value 0 (low). *Let* $I \in K_m^{MN}$ be an image of dimensions $M \times N$ and positive grayscale levels within [0, m], and W_i be a sub-image of I, such that the union of all of the sub-images reconstructs the original image I. The reconstruction of image I is defined by:

$$I = \bigcup_{1}^{s} w_i \tag{19}$$

One extension of SMNN is required for image compression: the auxiliary neuron is loaded with the sequential number of the corresponding MN from the second layer. The output of the MN is received by its corresponding outstar neuron, which when excited with a high input, outputs the value loaded during the training phase. Thus, the network effectively indicates the winner MN that has learned or recognized the input pattern (which in this particular application refers to a W_i window). This output must be preserved, mapping which W_i window a MN can reproduce. This is the key for decoding the compressed image; we use a mapping between a window and the neuron that has "learned" this window. In this way, the *nth* neuron's weights are used in order to reproduce the windows associated with it. This mapping is defined by

$$I^r = \{(k,n)_1, \ldots, (k,n)_i \mid \{k \in K, n \in S\}\}, \tag{20}$$

where S is the set of w_i sub-images, regarding image I. The value of k means the *kth* MN and $(k,n)_i$ refers to the *ith* mapping between a MN and a sub-image (window). Figure 2 shows how an image or a component of the frame is processed by a SMNN.

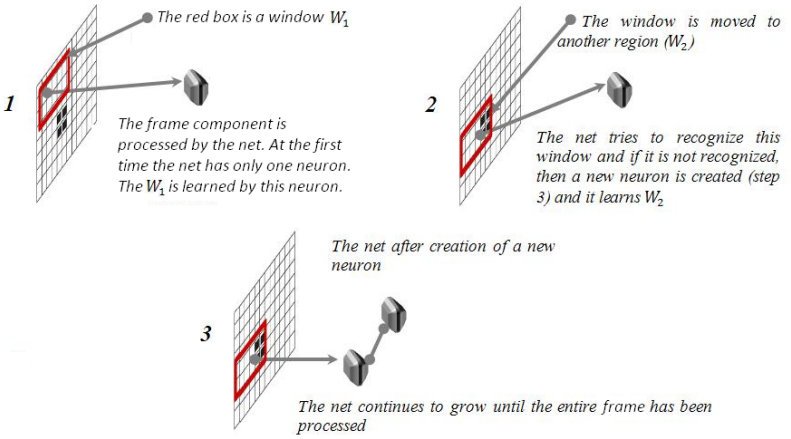

Fig. 2. Illustration of how image or frame is compressed by the SMNN

3.2 RGB Video Compression

For the compression of RGB video, each component of the stream is processed in an individual instance of SMNN. Consequently, at the end of the compression, we obtain three instances of SMNN. This process is depicted in Figure 3.

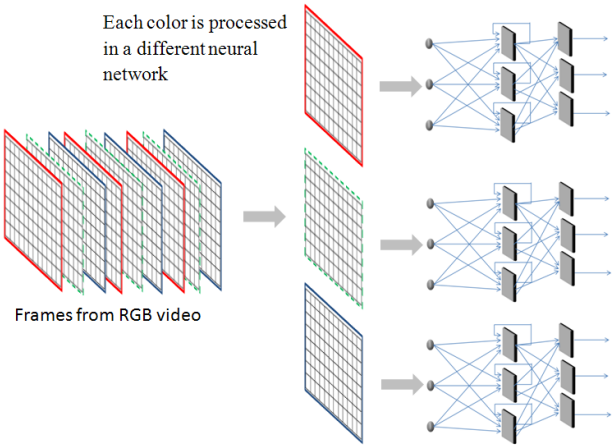

Fig. 3. RGB video compression by SMNN

The use of SMNN for video stream compression does not require any adaptation of its extended proposition for grayscale image compression. Each frame is split into a set of w_i sub-images. Then, for each set, we create a I^r map, according to equation (20). Finally, an entire component stream is encoded according to:

$$\bigcup_{i=1}^{N} I_i^r \,, \tag{21}$$

where N is the length of the video stream measured in frames. In the next section we present the SMNN results related to the compression of grayscale images and RGB video.

4 Computer Simulations and Discussion

Table 1 presents the results for the compression of Figure 2(G), which is a grayscale image with 320X240 pixels and 8 bits per pixel (bpp), totaling 76.800 bytes without compression.

Table 1. Results obtained by applying 3 image compression methods to figure 2(G)

Image	Format	Image Size MxN	Size (bytes)	Compression ratio C_R	Fidelity Criteria eRMS
A	PNG	320x240	77.279	(0,99)	2.6
B	PNG	320x240	54.858	1.39	8.4
C	NMC	320x240	72.040	1.07	0.3
D	NMC	320x240	30.146	2.55	4.9
E	JPG	320x240	41.684	1.84	3.6
F	JPG	320x240	2.468	31.11	27.5

Measurements of $eRMS$ and C_R are always calculated in relation to image 2(G). In Table 1, JPG refers to the Joint Photographic Experts Group format, PNG refers to the portable network graphics and NMC refers to the neural morphological compression method, produced by SMNN. Each method was used to produced images with the highest and lowest compression levels possible. To evaluate the fidelity criteria, we use the root mean square error (eRMS) for an objective evaluation of the images in Figure 2(A-G).

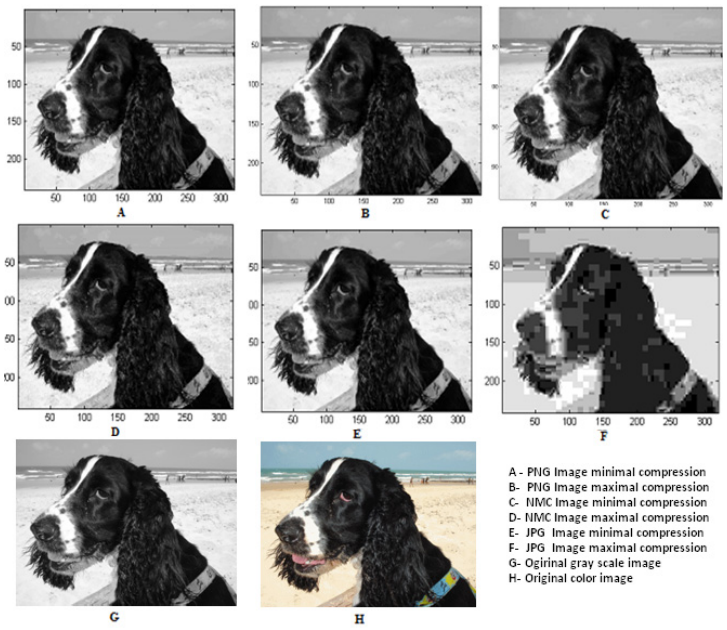

Fig. 4. Dog Lisbela in different images formats obtained with 3 compression methods

The compression ratio estimation listed in Table 1 were obtained in accordance with equation (22), in which s_1 refers to original image size in bytes, s_2 refers to compressed image size in bytes, and *eRMS* are defined according to equation (23).

$$C_{R=}\frac{s_1}{s_2} \tag{22}$$

$$e_{rms=}\left[\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f_2(x,y)-f_1(x,y)]^2\right]^{1/2} \tag{23}$$

Figure 5(A) depicts the results for compression with a window size = 3 and the following variations in the SMNN's parameters: *T* from 0.7 to 1.0, and *F* from 5 to 10. Note that in charts (A) and (B), the value of *F* was normalized. In chart (B), we can see the results for a window size = 3 and the same variation in the SMNN's parameters as seen in chart (A).

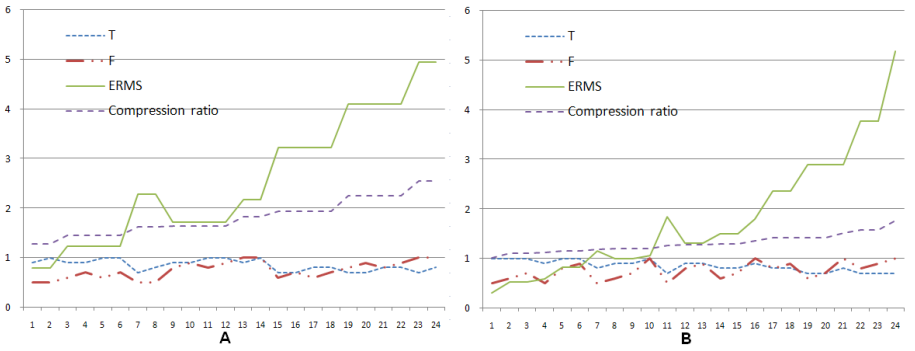

Fig. 5. Results for compression of grayscale image with variations of SMNN's parameters

For RGB video compression we used the "foreman" stream [13]. This test video was obtained by converting a CIF video to the RGB color space, sampling with 10 frames. The results are presented for 8×8 and 16×16 window sizes. In Figure 5 we can see samples of frame ten compressed with variations of SMNN's parameters.

Fig. 6. Frame ten compressed with various parameters.

Table 2. Samples of results obtained by applying SMNN to 10th frame of *Foreman* video

Frame	Window Size MxN	F	t	Compression ratio C_R	Objective Quality $PSNR$ (db)
A	8×8	10	0.90	1.296	45.47
B	8×8	20	0.90	2.357	34.33
C	8×8	30	0.90	3.698	30.25
D	16×16	10	0.90	1.162	40.31
E	16×16	20	0.90	1.690	34.20
F	16×16	30	0.90	2.227	30.59

The foreman video was successfully compressed by a SMNN, as we can see in table 2, for a windows size 8×8, with a low tolerance to variations in the pixel values (F = 10) and a pixel-error threshold setting of t=0.90, resulting in a good compression ratio (3.69) and an acceptable PSNR level. Note that these results refer to the 10th frame only. To evaluate the fidelity criteria for the compressed images and video we utilize the peak signal-to-noise ratio (PSNR) for an objective evaluation according to equation:

$$PSNR = \quad 10.\log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{24}$$

We extended the investigation of RGB video compression by compressing the first 100 frames of the Foreman video, and compared the results of SMNN with results from other well-known compression techniques. Table 3 shows these results. Figure 6 shows the PSNR and CR evolution, frame-by-frame, throughout the compression of the first 100 frames of the Foreman video, while Figure 7 shows the growth in the number of neurons during this compression.

In Table 3 the results for HEO-II refers to [14], KAMINSKY and JM9.5 refer to [15], and FS and ANEA refer to [16]. NMC1 refers to the results for SMNN with t=0.6, F=5.0, and a window with 8×8 pixels size, and NMC2 refers to results for SMNN with t=0.8, F=20.0, and a window with 4×4 pixels size.

Table 3. Numerical results obtained by applying SMNN to first 100 frames of *Foreman* video.

Technique	Requires complex pre-processing?	PSNR	Bit rate (bits/pixel)	Compression ratio
HEO-II-100F	(yes) H264/AVC	NA	0.660	1.51
KAMINSKY-100F	(yes) H264/AVC	35.85	0.0702	14.282
JM9.5-100F	(yes) H264/AVC	35.93	0.0705	14.182
FS	(yes) H264/AVC	36.33	0.384	2.60
ANEA	(yes) H264/AVC	36.29	0.543	1.84
NMC1	(none)	34.30	0.380	2.63
NMC2	(none)	27,93	0.104	9,59

In Figure 7 the sub-images refer to the number of elements in I_i^r, measured at the 100th frame (equation (21)).

As we can see in Tables 1 and 3, SMNN gave good results demonstrating that the network is capable of RGB video compression. Note that SMNN does not require pre-processing and all of results shown in this paper refer to the data without any secondary compression. Saving SMNN results to a hard-disk using trivial data compression can improve the final compression rates.

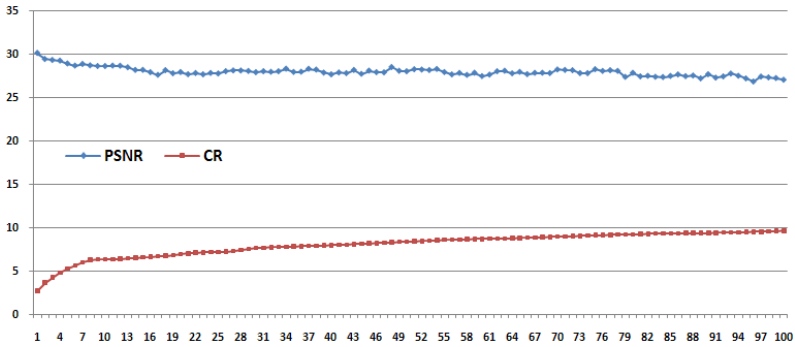

Fig. 7. Frame-by-frame evolution during compression of first 100 frames

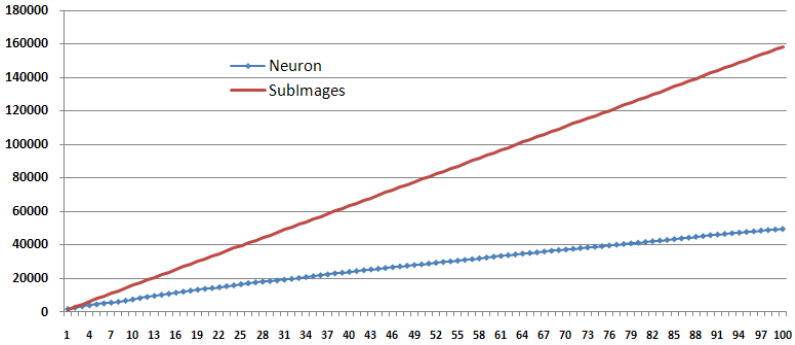

Fig. 8. Growth in the numbers of neurons during compression

5 Conclusion

This investigation and the detailed results for SMNN demonstrated that it is practical for RGB video and grayscale image compression and capable of producing results comparable to well-known methods. The reconstruction of the compressed image essentially occurs through data translation from neuron's weights to the respective *windows*, without requiring additional mathematical operations.

References

1. Winkler, S., van den Branden Lambrecht, C.J., Kunt, M.: Vision Models and Applications to Image and Video Processing, p. 209. Springer (2001)
2. Reddy, et al.: Image Compression and Reconstruction Using a New Approach by Artificial Neural Network. International Journal of Image Processing (IJIP) 6(2), 68–85 (2012)
3. Cramer, C., Gelenbe, E., Bakircloglu, H.: Low Bit-rate Video Compression with Neural Networks and Temporal Subsampling. Proceedings of the IEEE 84(10), 1529–1543 (1996)

4. Vaddella, R.P.V., Rama, K.: Artificial Neural Networks for Compression of Digital images: A Review. International Journal of Reviews in Computing, 75–82 (2010)
5. Singh, M.P., Arya, K.V., Sharma, K.: Video Compression Using Self-Organizing Map and Pattern Storage Using Hopfield Neural Network. In: International Conference on Industrial and Information Systems (ICIIS), December 28-31, pp. 272–278 (2009)
6. García-Rodríguez, J., Domínguez, E., Angelopoulou, A., Psarrou, A., Mora-Gimeno, F.J., Orts, S., García-Chamizo, J.M.: Video and Image Processing with Self-Organizing Neural Networks. In: Cabestany, J., Rojas, I., Joya, G. (eds.) IWANN 2011, Part II. LNCS, vol. 6692, pp. 98–104. Springer, Heidelberg (2011)
7. Khashman, A.: Neural Networks Arbitration for Optimum DCT Image Compression. In: IEEE Eurocon (2007)
8. Banon, G.J.F.: Characterization of Translation Invariant Elementary Morphological Operators Between Gray-level Images. INPE, São José dos Campos, SP, Brasil (1995)
9. Banon, G.J.F., Faria, S.D.: Morphological Approach for Template Matching. In: Brazilian Symposium on Computer Graphics and Image Processing Proceedings. IEEE Computer Society (1997)
10. Faria, S.D.: Uma abordagem morfológica para casamento de padrões, Master Tesis, National Institute for Space Research, INPE-6346-RDI/597 (1997)
11. Silva, F.A.F.S., Banon, G.J.F.: Rede morfológica não supervisionada (RMNS). In: IV Brazilian Conference on Neural Networks, pp. 400–405 (1999)
12. Banon, G.J.F., Barrera, J.: Decomposition of Mappings Between Complete Lattices by Mathematical Morphology – Part I: General Lattices. Signal Processing 30(3), 299–327 (1993)
13. Foreman, Video stream for tests, `http://trace.eas.asu.edu/yuv/`
14. Heo, J., Ho, Y.-S.: Efficient Differential Pixel Value Coding in CABAC for H.264/AVC Lossless Video Compression. Circuits, Systems and Signal Processing 31(2), 813–825 (2012)
15. Kaminsky, E., Grois, D., Hadar, O.: Dynamic Computational Complexity and Bit Allocation for Optimizing H.264/AVC Video Compression. Journal of Visual Communication and Image Representation 19(1), 56–74 (2008)
16. Saha, A., Mukherjee, J., Sural, S.: A Neighborhood Elimination Approach for Block Matching in Motion Estimation. Signal Processing, Image Communication 26(8), 438–454 (2011)