SCIENCE CHINA
Information Sciences

• Supplementary File •

Quantized and adaptive memristor based CNN (QA-mCNN) for image processing

HU XiaoFang[1,3], SHI WenQiang[2], ZHOU Yue[4], TANG HongAn[5] & DUAN ShuKai[1*]

[1] *College of Artificial Intelligence, Southwest University, Chongqing* 400715*, China;*
[2] *College of Computer and Information Science, Southwest University, Chongqing* 400715*, China;*
[3] *School of Mathematics and Statistics, Southwest University, Chongqing* 400715*, China;*
[4] *College of Electronics and Information Engineering, Southwest University, Chongqing* 400715*, China;*
[5] *School of Artificial Intelligence, Chongqing University of Technology, Chongqing* 401135*, China*

Appendix A Proof of flux-controlled memristor model

As we can know from the dynamics and output function of a cell $c(i,j)$ in eq. (1) and (2), the current through the state memristor can be got:

$$m(x_{ij}(t)) = \frac{v_m}{M(t)} = \frac{x_{ij}(t)}{M(t)} \tag{A1}$$

where $M(t)$ and v_m represent the memristance and voltage of the state memristor, respectively. According to the typical HP TiO_2-based memristor model [1], the memristance is given by:

$$M(t) = R_{OFF} + (R_{ON} - R_{OFF})\frac{W(t)}{D} \tag{A2}$$

where R_{OFF} and R_{ON} denote the maximum and minimum memristance values, respectively. W(t) denotes the width of the doping layer (TiO_{2-x}), and D denotes the thickness of TiO_2 film [1].

According to the linear drift model, we have:

$$\frac{dW(t)}{dt} = \frac{\mu_v R_{ON}}{D} i(t) \tag{A3}$$

μ_v represents the average mobility of oxygen vacancies, and:

$$W(t) = \frac{\mu_v R_{ON}}{D}\phi(t) + W(0) \tag{A4}$$

In ref [2], we have:

$$k = \frac{(R_{ON} - R_{OFF})\mu_v R_{ON}}{D^2}, \phi(t) \in [\frac{R_{OFF} - M(0)}{k}, \frac{R_{ON} - M(0)}{k}] \tag{A5}$$

the following equation can be derived:

$$M(t) = M(0) + k\phi(t) \tag{A6}$$

where $\phi(t)$ and $M(0)$ denote the flux and initial memristance of the memristor. And then the charge-controlled model can be given :

$$M(t) = \begin{cases} R_{OFF}, & \phi(t) < c_1 \\ M(0) + k\phi(t), & c_1 \leqslant \phi(t) < c_2 \\ R_{ON}, & \phi(t) \geqslant c_2 \end{cases} \tag{A7}$$

where, $c_1 = (R_{OFF}^2 - M^2(0))/2k, c_2 = (R_{ON}^2 - M^2(0))/2k$. And according to the relationship between charge and magnetic flux, the magnetic flux-control model of memristor in (A8) can be obtained.

$$M(t) = \begin{cases} R_{OFF}, & \varphi(t) < c_1 \\ \sqrt{2k\varphi(t) + M^2(0)}, & c_1 \leqslant \varphi(t) < c_2 \\ R_{ON}, & \varphi(t) \geqslant c_2 \end{cases} \tag{A8}$$

* Corresponding author (email: duansk@swu.edu.cn)

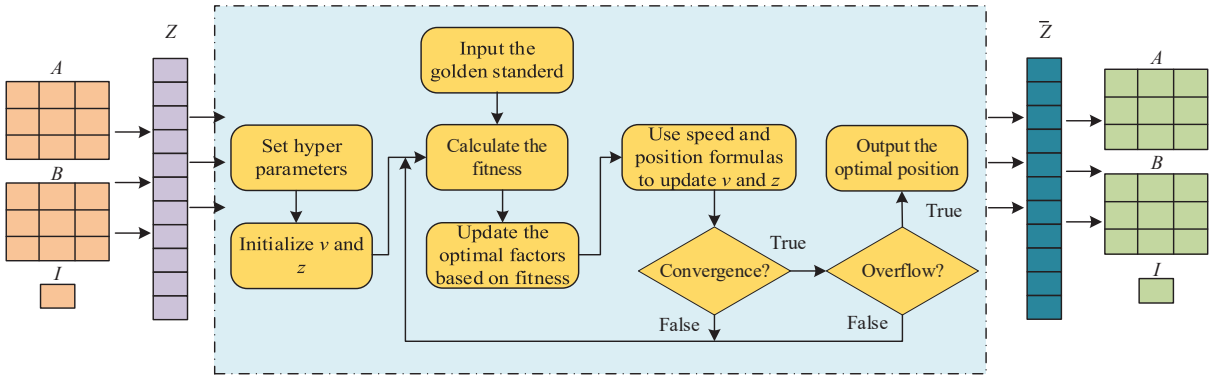


Figure B1 Adaptive template generation based on PSO algorithm

Appendix B The process of adaptive template generation

In this section, the hardware-friendly adaptive template design based on optimization algorithm and quantization will be described in detail.

mPSO is a heuristic algorithm with a fast convergence speed and insensitivity to the population number. This method can find the optimal solution by random initialization and step-by-step iterative approximation, where the approximation level can be evaluated by a fitness function. In general, the variation trend of the fitness is a convergent curve.

As described in state function of mCNN, the inputs benefit is decided by the control template (B), feedback template (A) and bias (I). At the condition of $r = 1$, these templates can be expressed by:

$$A = \begin{bmatrix} a_{i-1,j-1} & a_{i-1,j} & a_{i-1,j+1} \\ a_{i,j-1} & a_{i,j} & a_{i,j+1} \\ a_{i+1,j-1} & a_{i+1,j} & a_{i+1,j+1} \end{bmatrix}, B = \begin{bmatrix} b_{i-1,j-1} & b_{i-1,j} & a_{i-1,j+1} \\ b_{i,j-1} & b_{i,j} & b_{i,j+1} \\ b_{i+1,j-1} & b_{i+1,j} & b_{i+1,j+1} \end{bmatrix}, I = I \tag{B1}$$

Generally, the templates A and B are symmetric, so the parameters to be determined can be reduced by nearly half, that is, about 11. These templates can be rewritten as

$$A = \begin{bmatrix} Z_2 & Z_3 & Z_4 \\ Z_5 & Z_1 & Z_3 \\ Z_4 & Z_5 & Z_2 \end{bmatrix}, B = \begin{bmatrix} Z_7 & Z_8 & Z_9 \\ Z_{10} & Z_6 & Z_8 \\ Z_9 & Z_{10} & Z_7 \end{bmatrix}, I = Z_{11} \tag{B2}$$

which can be abbreviated as:

$$A = [Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7, Z_8, Z_9, Z_{10}, Z_{11}] \tag{B3}$$

Next, these parameters can be generated and optimized by mPSO algorithm. Let z denotes the position matrix, indicating the current position of the particle swarm, and v represents the velocity matrix, indicating the convergence speed of the particle swarm. At first, z and v are randomly initialized within certain constraints, respectively:

$$|z(i,j)| \leqslant 8, 0 < i < m; 0 < j < n \tag{B4}$$

$$|v(i,j)| \leqslant 1, 0 < i < m; 0 < j < n \tag{B5}$$

where the constant m represents the total number of the particle swarms, n=11 denotes the search dimension or the number of independent variables, parameters i and j represent the row and column in the matrix, respectively. It is important to note that the restriction of v should be moderate correspondly.

Supposing the input size of an image is $x \times l$, then the fitness function based on characteristics of mCNN can be written as:

$$cost = \left(\frac{\sum_{i=1}^{s}\sum_{j=1}^{l}|y(i,j) - \hat{y}(i,j)|}{\sum_{i=1}^{s}\sum_{j=1}^{l}|y(i,j)|}\right)^2 \tag{B6}$$

where $y(i,j)$ denotes the standard result of the pixel at (i,j) of an image. $\hat{y}(i,j)$ denotes the processing result under the current iteration using the optimized templates. This fitness function is a concave function with only one optimal solution, which can guarantee the calculation to be positive and the mPSO algorithm can converge to the global optimum.

The framework of the adaptive template generated by PSO algorithm is shown in Fig. B1. First, the template parameters are splinted as independent variables z. Then, the speed v and position z are initialized and participated in the iteration. During the iteration processes, the fitness is calculated fleetly to update the optimal position z. As shown in the right-hand side of Fig. B1. In the end, when the fitness tends to converge, the optimal position will be found, and the adaptive template will be generated after proper reconstruction.

Appendix C Experimental results and analysis

In this section, two kinds of applications on image processing has been analyzed to demonstrate the performance of the proposed scheme. The experimental platform is MATLAB2016a. Before performing the application experiment, color images need to be converted to grayscale images and contrast enhancement is also needed especially for image segmentation.

CNNs have shown excellent performance in edge extraction [3,4]. However, traditional edge extraction templates lack adaptability and flexibility, hence they are not ideal for dealing with complex images. In addition, although some non-linear and/or adaptive templates like the adaptive bionic templates [5] can provide better performance, they all stay in software level because of the complicated hardware implementation.

To demonstrate the effectiveness and superior performance of QA-mCNN, a series of simulations and comparisons with four traditional edge extraction operators and three kinds of CNNs will be presented. In the edge extraction QA-mCNN, the template's layer S is set to be 1 and the bit-width of template quantization $b - a$ is 3.

(a) (b) (c)

Figure C1 Edge extraction of adaptive mCNN. (a) Original image, (b) Result of PSO mCNN (c) Result of QA-mCNN

Fig. C1 illustrates the edge extraction results of the image Lena (Fig. C1(a)) using the mCNN with adaptive templates based on PSO algorithm (PSO mCNN) (Fig. C1(b)) and QA-mCNN (Fig. C1(c)). It can be seen that the PSO mCNN can extract the edges more completely and restores the details of hair, eyes and background, which shows the superior performance of the generated adaptive templates. QA-mCNN also achieves satisfactory results. Compared with the edge extraction results shown in Fig. C1, Fig. C2 shows the results from other methods. It can be seen that Robert operator, Prewitt operator and Sobel operator can hardly extract the edges completely. Although Canny operator can extract many more edges, a lot of distortions are caused. In addition, the adaptive bionic mCNN [5] and the standard CNN have shown superior performance, especially in the details as in Fig. C2(f) and (a).

(a) (b) (c)

(d) (e) (f)

Figure C2 Comparison results of edge extraction from (a) Standard CNN (with fixed templates), (b) Sobel Operator, (c) Robert Operator, (d) Prewitt Operator, (e) Canny Operator, and (f) the adaptive bionic mCNN [5].

In a summary, the edges extracted by PSO mCNN, QA-mCNN, the adaptive bionic mCNN and the standard CNN have higher accuracy and completeness. QA-mCNN also captures advantages in hardware and computational complexity at the cost of little quantization loss.

Moreover, to quantitatively and objectively analyse the effect of edge extraction, a performance evaluation schema: FOM (Figure of Merit) [6] that has been widely used in image analysis is utilized. FOM, a widely used performance evaluation schema in image analysis, is utilized as defined by:

$$FOM = \frac{1}{max(N_I, N_T)} \sum_{i=1}^{N_T} \frac{1}{1 + \alpha d_i^2} \tag{C1}$$

where N_I denotes the total number of pixels in the golden standard image, N_T denotes the total pixels of the edge image that are detected by QA-mCNN. α is a constant also called the compensation coefficient and used to compensate for the offset edge of image (the value is 1/9). d_i represents the shortest distance between the actually detected edge pixel and the standard pixel in the same position. The constraint is $FOM \in [0, 1]$.

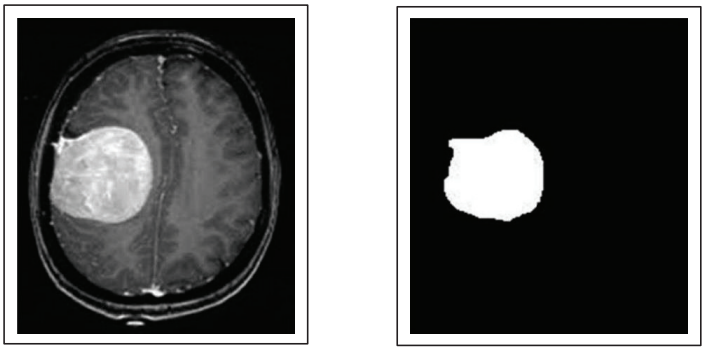
Table C1 summarizes the comparative data obtained from the above-mentioned edge extraction methods. It can be seen that compared with the adaptive bionic mCNN [5] and the standard mCNN, the proposed QA-mCNN and PSO mCNN have better FOM values on edge extraction. Furthermore, QA-mCNN also possesses

Table C1 FOM of edge extraction

Method	Sobel	Robert	Prewitt	Canny	[5]	PSO mCNN	Standard CNN	QA-mCNN
FOM	0.2543	0.2335	0.2517	0.4393	0.9568	0.9792	0.9234	0.9697

advantages of hardware-friendly property and adaptivity. On the contrary, FOM values of other methods (Sobel, Robert, Prewitt, Canny) are too low to extract the image edges effectively.

In recent years, image segmentation has played an important role in various fields, such as object detection in industrial field, face recognition in daily life, and tumor segmentation in medical field. Therefore, image segmentation is taken as another important example to illustrate the performance of QA-mCNN by a series of simulations and comparisons among three kinds of CNNs. Fig. C3 presents an original image randomly selected from DDSM in (a) and it's gold standard in (b).



(a) (b)

Figure C3 Example of image segmentation. (a) Original image and (b) Gold standard image

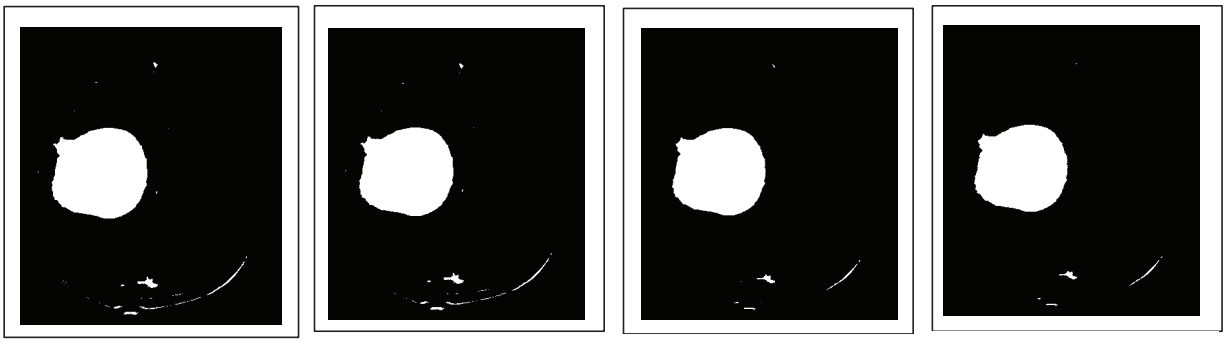
First, the bit-width $(b-a)$ and layer (S) of the image segmentation QA-mCNN are set to $b-a=3$, $S=3$, respectively. Then execute the segmentation of the image in Fig.C3 based on the designed adaptive templates and multilayer QA-mCNN. The simulation results are shown in Fig. C4. It can be seen that when $S=3$ (Fig. C4(d)) the tumor image has been segmented with high accuracy and integrity. And compared with QA-mCNN, the image segmented by PSO mCNN (Fig. C4(a)) has lower accuracy, because PSO mCNN just has 1-layer of linear template, although it is not quantized.

Additionally, to intuitively illustrate the advantages of QA-mCNN, some comparisons with the standard CNN and the quantized CNN proposed in Ref. [7] are made as shown in Fig. C5. It can be seen that the standard CNN (Fig. C5(a)) with fixed templates cannot segment the tumor image, while by using the quantized CNN, when $S=3$ (Fig. C5(d)), the target image can be segmented much better, but the segmentation accuracy is much lower than QA-mCNN, as shown in Fig. C4(d).

According to the comparison between the Fig. C4 and Fig. C5, it can be found that QA-mCNN has higher image segmentation accuracy. At the same time, QA-mCNN also has the advantages of low computation complexity, adaptivity and easy hardware implementation.

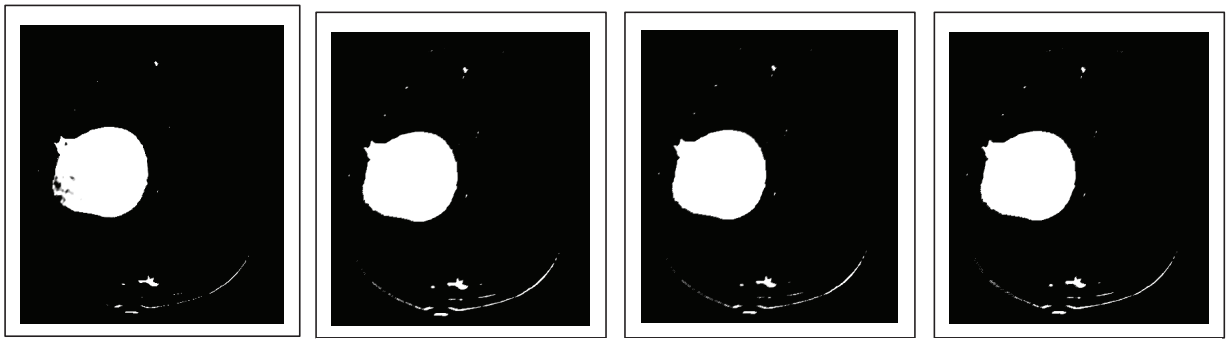
To be more objectively, the commonly used evaluation indices such as Dice ratio [8], Ground Truth (GT), Over Segmentation (OS), Under Segmentation (US) and Pixel Accuracy (PA) [9] have been employed. Based on the corresponding calculations, Table C2 summarizes the comparative data between our schemes and other methods. It can be seen that QA-mCNN exhibits satisfactory performance, where the Dice ratio and GT keep in a stable range. When S=3, QA-mCNN can stabilize US at a low level and effectively reduce OS, solving the NP-hard problems between over-segmentation and under-segmentation. In addition, the pixel accuracy is continuously improved with the number of template's layers increasing.

Based on the simulation and analysis, QA-mCNN has exhibited high accuracy and low computational complexity. The main reasons include several aspects as follows. The quantization processes is designed based on multilevel memristor technology, which reduces the accuracy loss. INQ can further compensate the loss caused by quantization. The templates of each layer are optimized by the previous templates, eliminating some randomness. Besides, since QA-mCNN is constructed based on memristor crossbar array, it also possesses unique hardware implementation advantage.



(a) (b) (c) (d)

Figure C4 Image segmentation results of the proposed methods. (a) PSO mCNN, (b) QA-mCNN b-a=3 S=1, (c) QA-mCNN b-a=3 S=2, (d) QA-mCNN b-a=3 S=3.



(a) (b) (c) (d)

Figure C5 Image segmentation results of other methods. (a) Standard CNN, (b) Quantized CNN proposed in [8] b-a=3 S=1, (c) Quantized CNN proposed in [8] b-a=3 S=2, (d) Quantized CNN proposed in [8] b-a=3 S=3.

Table C2 Indicators of segmentation results

Indicators		Dice	GT	OS	US	PA
Standard CNN		0.93684	0.91633	0.09718	0.00458	0.97604
PSO mCNN		0.93198	0.87348	0.11608	0.00478	0.9659
[8]	b-a=3, S=1	0.92917	0.94077	0.085891	0.039736	0.96527
[8]	b-a=3, S=2	0.93844	0.96859	0.096348	0.023298	0.86197
[8]	b-a=3, S=3	0.92725	0.88354	0.11012	0.005319	0.89384
$QA-mCNN$	b-a=3, S=1	0.94391	0.88354	0.081177	0.014732	0.89546
$QA-mCNN$	b-a=3, S=2	0.96244	0.92998	0.078305	0.00000	0.96005
$QA-mCNN$	b-a=3, S=3	0.94623	0.92606	0.064418	0.00000	0.99564

Table D1 Variation analysis of edge extraction

Method	QA-mCNN	QA-mCNN with 5% template variation	QA-mCNN with 15% template variation
FOM	0.9535	0.9447	0.9224

Appendix D Variations analysis of template's weights

The variation of template's weight represented by memristors may result from the fabrication variation of memristor devices, weight programming inaccuracy, circuit noise, and so on. As far as we know, to a certain extent, the quantization process in INQ can improve the fault tolerance of the mCNN. SO, in order to facilitate analysis, the variation of memristor weights is assumed to follow the law of Gaussian distribution with mean of zero. It is supported by experimental data that when $\sigma = 0.1$, the difference between the desired and the actual weight is up to 15%. Similarly, when $\sigma = 0.025$, the difference is about 5%.

Fig. D1 shows the influence of single layer QA-mCNN variation on edge extraction of Lena. Figs. D1 (a) and (b) represent the input image and the outputs processed by QA-mCNN without template variations, respectively. When considering the templates with 5% and 15% variations on the single layer of the QA-mCNN, the corresponding outputs can be obtained as shown in Fig. D1(c) and (d), separately.

Quantitative indicator FOM of QA-mCNN under these conditions is measured as shown in Table D1. It can be seen that even with bigger template variations (15%), the QA-mCNN can still extract the image's edges successfully (Fig. D1(d)) and have high FOM value.

The variation analysis of QA-mCNN with multilayer templates is more complicated. For instance, in the image segmentation, the variation is considered not only in layer-by-layer-same but also in layer-by-layer-different memristor templates. Therefore, four kinds of schema have been considered according to the different variation (variation with 5% and 15%) and variation distribution (layer-by-layer same and layer-by-layer different) in this paper, and the experimental

(a) (b) (c) (d)

Figure D1 Comparison of the outputs of edge detection by QA-mCNN without or with weight variations. (a) Input image. (b) Output of edge detection without weight variations. (c) Output of edge detection with 5% template variations and (d) Output of edge detection with 15% template variations.

Table D2 Variation analysis of image segmentation

Indicators	Dice	GT	OS	US	PA
QA-mCNN	0.95187	0.92432	0.07773	0.0069911	0.99215
5% layer-by layer-same template variations	0.95099	0.9254	0.077211	0.0077915	0.99296
5% layer-by-layer-different template variations.	0.9553	0.91538	0.084645	0.0077915	0.99093
15% layer-by-layer-same template variations	0.9503	0.91172	0.077211	0.0089897	0.99215
15% layer-by-layer-different template variations	0.95525	0.91193	0.07773	0.0085241	0.9914

results in image segmentation are shown in Fig. D2. It can be seen that small variation (5%) in templates does not significantly affect the outputs of image segmentation, as shown in fig. D2(c) and fig. D2(d). Even with bigger variation (15%) of layer-by-layer same/different in templates, QA-mCNN still segments the tumor position successfully, as shown in fig. D2(e) and fig. D2(f).

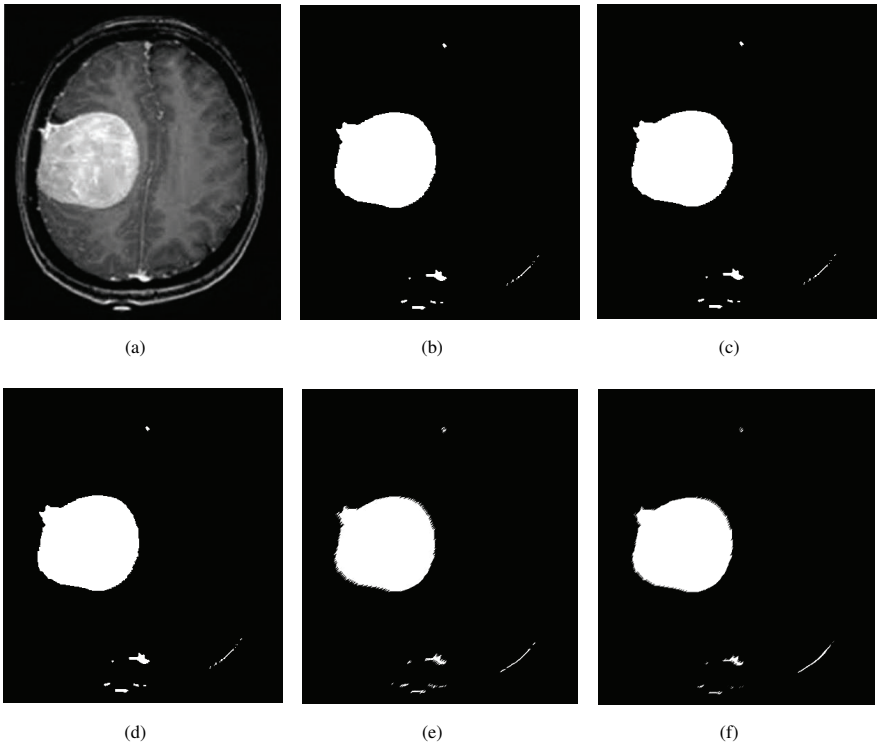


Figure D2 Comparison of the outputs of image segmentation via a 3-layer QA-mCNN without or with weight variations. (a) Input image. (b) Output of image segmentation without weight variations. The outputs of image segmentation with (c) 5% layer-by layer-same template variations. (d) 5% layer-by-layer-different template variations. (e) 15% layer-by-layer-same template variations, and (f) 15% layer-by-layer-different template variations, respectively.

The objective analysis based on quantitative indexes is summarized in Table D2, which shows the consistency with the results of Fig.D2. In particular, even using the QA-mCNN with 15% layer-by-layer-different template variations, the Dice, GT and PA of the segmented image still remain a stable range and have a high value, OS and US stay at a low level and keep in a stable range too.

So, QA-mCNN has effective fault tolerance in image processing, even if the variation can affect the accuracy of the templates. In summary, variations of the memristor templates may be unavoidable, the quantization process in INQ can, however, overcome this kind of change to some extent, which can improve the fault tolerance and robustness of QA-mCNN.

References

1 Strukov D B, Snider G S, Stewart D R, et al. The missing memristor found. Nature, 2008, 452: 80-83.
2 Wang L D, Drakakis E, et al. Memristor model and its application for chaos generation. International Journal of bifurcation and chaos, 2012, 22: 1250205-.
3 Ren L, Hao X, Xie X. A new algorithm for CNN template design based genetic algorithm. Journal of Shandong Institute of Technology, 2001, 15: 48-51
4 Hu X, Feng G, Liu L, Duan S. Composite Characteristics of Memristor Series and Parallel Circuits. International Journal of Bifurcation and Chaos, 2015, 25: 1530019.
5 Yang T, Duan S, Wang L, et al. Edge extraction of color image based on improved memristor cell neural network. Science China Information Science, 2017, 7: 57-71.
6 Yu Y, Chang C. A new edge detection approach based on image context analysis. Image Vision Compute, 2006, 24: 1090-1102.
7 Liu Z, Zhuo C, Xu X. Efficient segmentation method using quantised and non-linear CeNN for breast tumor classification. Electronics Letters, 2018. 54: 737-738.
8 Dice L R, Measures of the Amount of Ecologic Association between Species. Ecology, 1945, 26: 297-302
9 Chang H H, Zhuang A H, Valentino D J, et al. Performance measure characterization for evaluating neuroimage segmentation algorithms. NeuroImage, 2009, 47: 122-135.