ARTICLES
https://doi.org/10.1038/s41928-017-0002-z
nature electronics

Analogue signal and image processing with large memristor crossbars

Can Li[1], Miao Hu[2,5], Yunning Li[1], Hao Jiang[1], Ning Ge[3], Eric Montgomery[2], Jiaming Zhang[2], Wenhao Song[1], Noraica Dávila[2], Catherine E. Graves[2], Zhiyong Li[2], John Paul Strachan[2]*, Peng Lin[1], Zhongrui Wang[1], Mark Barnell[4], Qing Wu[4], R. Stanley Williams[2], J. Joshua Yang[1]* and Qiangfei Xia[1]*

Memristor crossbars offer reconfigurable non-volatile resistance states and could remove the speed and energy efficiency bottleneck in vector-matrix multiplication, a core computing task in signal and image processing. Using such systems to multiply an analogue-voltage-amplitude-vector by an analogue-conductance-matrix at a reasonably large scale has, however, proved challenging due to difficulties in device engineering and array integration. Here we show that reconfigurable memristor crossbars composed of hafnium oxide memristors on top of metal-oxide-semiconductor transistors are capable of analogue vector-matrix multiplication with array sizes of up to 128 × 64 cells. Our output precision (5–8 bits, depending on the array size) is the result of high device yield (99.8%) and the multilevel, stable states of the memristors, while the linear device current–voltage characteristics and low wire resistance between cells leads to high accuracy. With the large memristor crossbars, we demonstrate signal processing, image compression and convolutional filtering, which are expected to be important applications in the development of the Internet of Things (IoT) and edge computing.

Improvements in the energy consumption and throughput of digital processors are reaching a plateau, as complementary metal-oxide-semiconductor transistor (CMOS) technology approaches the end of process scaling[1,2]. This issue impacts the power requirements of large data centres, or the Cloud, and can also limit the effective deployment of sensors and actuators for the Internet of Things (IoT)[1,3] because of limited communication bandwidth and the cost of data transmission. There is no way to transmit and store all the data being gathered now for central analysis, and this challenge is expected to grow with the orders of magnitude more devices expected with the development of the IoT. The result is that the edge of the network will need sufficient intelligence[4] to pre-process data in place and transmit only the most important information to the Cloud. This edge computation will have to be extremely power efficient, as it may depend only on the energy that it can scavenge from its environment. Thus, new computational devices and approaches are critical, especially those that can interface directly to the analogue output of embedded sensors to filter, analyse, compress, encode and possibly encrypt data before transmittal.

Many of these operations can be expressed as a vector-matrix multiplication (VMM), which in principle can be performed in the analogue domain by a memristor crossbar array[5–10] using Ohm's law for multiplication and Kirchhoff's current law for summation[11–30] (Fig. 1a). Such VMMs are being developed as accelerators for inference on deep neural networks[31–35], but may also be used as reconfigurable analogue processors for edge computing. A vector of voltage outputs from a sensor can be applied directly to the rows of a memristor crossbar, in which the values of the appropriate matrix elements have been stored as the conductance of the cells. The currents that appear on the columns of the array in real time represent the output vector of the multiplication if the series resistance of the interconnection wires is negligible compared with the memristor resistances. To read out the results in parallel, the current signal from each column is converted to a voltage signal through a transimpedance amplifier (TIA), which also serves as a virtual ground.

So far, demonstrations of this concept have been limited to binary signal input and/or binary matrix weights[14–16]. Recently, pulse width, instead of amplitude, was used to represent the analogue input signals[27–30], but this scheme requires more readout time and more complicated integrated circuits. Previous experimental demonstrations of an analogue-voltage-amplitude-vector by analogue-conductance-matrix product, to the best of our knowledge, have been limited to a 1 × 3 system[24–26], which is not strictly a VMM implementation. Here, we report completely analogue VMMs with adequate accuracy and high speed–energy efficiency that are based on up to 128 × 64 crossbars of hafnium oxide (HfO_2) memristors[36], and experimentally demonstrate the important IoT and network edge applications of signal spectrum analysis, image compression and convolutional filtering.

128 × 64 memristor crossbars

To precisely tune the conductance of each memristor in a crossbar, we monolithically integrated a memristor on top of a metal–oxide–semiconductor (MOS) transistor as an access device in each cell, which is known as the '1T1R' architecture. Compared with passive arrays that use highly nonlinear memristors[14,37–39] or discrete selector devices[40–43] to mitigate the sneak path current problem, the 1T1R scheme has a lower packing density (2.5 times the cell area). However, it allows us to independently access memristors with a linear current–voltage (I–V) relation in an array with the transistor gate control, so each memristor's conductance can be precisely tuned. Moreover, unlike passive arrays, a 1T1R crossbar enables accurate analogue VMM with linear I–V memristors that yield a good approximation to the scalar product of a vector component and matrix element. The transistors also take advantage of the maturity of the CMOS platform and hence

[1]Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, USA. [2]Hewlett Packard Labs, Hewlett Packard Enterprise, Palo Alto, CA, USA. [3]HP Labs, HP Inc., Palo Alto, CA, USA. [4]Air Force Research Laboratory, Information Directorate, Rome, NY, USA. Present address: [5]Department of Electrical and Computer Engineering, Binghamton University, Binghamton, NY, USA. *e-mail: john-paul.strachan@hpe.com; jjyang@umass.edu; qxia@umass.edu

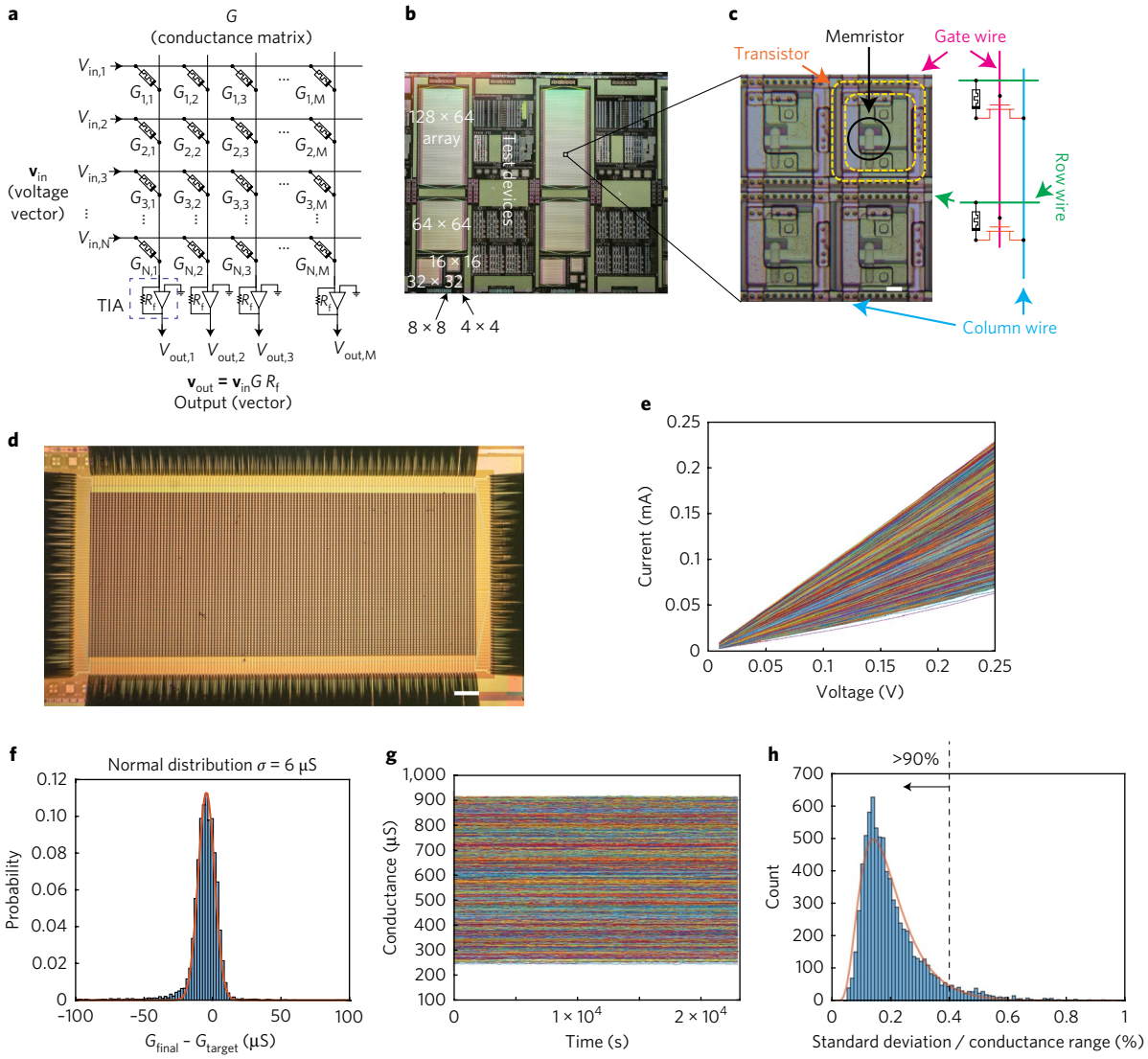

Fig. 1 | Data stored in a 128 × 64 1T1R memristor crossbar, demonstrating conductance state linearity, write precision and accuracy, and read stability and reproducibility. a, Schematic of the VMM operation. Multiplication is performed via Ohm's law, as the product of the voltage applied to a row and the conductance of a crosspoint cell yields a current injected into the column, while the currents on each column are summed according to Kirchhoff's current law. The total current from each column is converted to a voltage by a TIA, which also provides a virtual ground for the column wires. **b**, A 2 cm × 2 cm detail from a photograph showing two dies of 1T1R memristor crossbars, each of which contains array sizes from 4 × 4 to 128 × 64 cells, along with various test devices. **c**, Microscope image of four cells in a 1T1R array (scale bar, 10 μm). Crosses are memristors, and the transistors are ring-shaped. Inset: schematic showing how the memristors and transistors are connected into an array. **d**, Photograph of a probe card in contact with an operational 128 × 64 1T1R array (scale bar, 500 μm). **e**, Quasi-d.c. *I*–*V* curves for all the devices with different conductances, showing good *I*–*V* linearity over the selected conductance range. **f**, Histogram of the initial difference between the target and measured conductance written into a 128 × 64 array. A fit of the peak to a normal distribution yielded a standard deviation of 6 μS, with the peak maximum located at −5 μS. **g**, Room-temperature state retention and read disturb of the device states. The d.c. conductance states of all the devices were measured with a 0.2 V bias for 1,000 cycles, or a total of 6.4 h, showing no discernible drift in the plots. **h**, Histogram of the normalized standard deviation (s.d.), defined as the s.d. per conductance range (100–900 μS), for all measured states, which was fitted to a lognormal distribution. This shows that there are fluctuations during the read operation that can occasionally degrade the effective precision of an individual memristor, but 90% of the device states have a normalized s.d. less than 0.39%.

are attractive for applications in which packing density is not the most critical factor. In principle, depletion-mode transistors that are in the 'on' state at zero gate–source voltage can be used so that gate voltages on transistors are only needed for memristor array programming but not for normal VMM operations. We used n-type enhancement-mode transistors in this demonstration. The choice of transistor and its effect on leakage is discussed in Supplementary Note 1. Integration was conducted at UMass Amherst by building Ta/HfO_2/Pd (ref. [36]) memristors on top of a CMOS chip fabricated by a commercial vendor (see Methods for more details). Figure 1b shows part of the integrated chip consisting of 1T1R arrays with sizes ranging from 4 × 4 to 128 × 64. The detailed structure of some cells and the connection scheme are shown in Fig. 1c and Supplementary Fig. 1. The source wires of the transistors are rotated by 90° with respect to the source wire design for the 1T1R memories, so that when all the transistors are turned on, the array converts into a fully connected memristor crossbar.

The programming and computing were achieved with a custom-built testing system connected to the chip by a probe card (see Methods and Supplementary Fig. 2). Figure 1d shows a 128×64 array with probes touching the contact pads. With the 1T1R scheme the array size can be much larger than 128×64, but this array was chosen for the demonstration mainly because of the constraint of the maximum number of probes (388, as shown in Fig. 1d) available on the commercial probe card used for testing. With transistors as the access devices, we were able to program the conductance of nearly all of the memristors to an arbitrary value within a predefined conductance range (Supplementary Video 1). We wrote MATLAB scripts to control the resistance tuning by communicating with the testing system. With the Ta/HfO_2/Pd memristors, the I–V relation of the cells was linear once the conductance was larger than the quantum conductance (77.5 μS)[44,45], as shown in Fig. 1e for conductance ranging from 300 to 900 μS, an important feature for accurate analogue computing. Typical resistance switching curves are plotted in Supplementary Fig. 3.

Among the 8,192 devices in the 128×64 array, there were only three stuck 'on' and 15 stuck 'off' devices after programming, leading to a responsive device yield of 99.8%. A histogram of the writing error, defined as the initial difference between the target conductance value and the measured written value of the responsive memristors, is plotted in Fig. 1f (more data, including those from differently sized arrays, are shown in Supplementary Fig. 4). The peak of the writing error conformed to a normal distribution with a standard deviation σ of 6 μS when the writing tolerance was set to ±10 μS, and could be further reduced by defining a narrower tolerance in the MATLAB script and/or using a larger number of closed-loop programming iterations, at the expense of increased programming time. If, for the moment, we discount the tail of the distribution, which represents a small number of 'sticky' cells, and define the interval between states as $\pm\sigma$, we have effectively demonstrated more than 64 levels of conductance or 6 bits of digital precision over the conductance range 100–900 μS, which has been proven to be sufficient for many tasks in machine learning algorithms[13,15]. The accuracy error δG of the memristor programming operation is taken to be the median value of the writing error, which is −4.7 μS. To explore the read stability and reproducibility, we measured the conductance of the responsive 8,174 devices in the 128×64 array with 0.2 V read pulses for more than 6 h and did not see any detectable state drift (Fig. 1g). There were fluctuations in the read operations of individual cells, but these were small enough to have little impact on column current measurements summed over multiple memristors. These fluctuations, however, are a good indicator of the ultimate bit precision of the system. For example, 90% of device states have fluctuations within a 0.39% normalized standard deviation (Fig. 1g and Supplementary Fig. 5), indicating that the writing precision is 128 states or 7 bits in the conductance range 100–900 μS. The writing error and readout stability are not correlated with the selected conductance range (Supplementary Figs. 6 and 7), demonstrating the simplicity of making use of the multilevel conductance states. The device maintains the stable states at normal working temperatures (room temperature to 85 °C, Supplementary Fig. 8). The stable multilevel conductance states may be a result of the high migration barrier (measured value 1.55 eV)[36] for the Ta cations and O anions within a Ta-rich conductance channel formed in the HfO_2 matrix for the Ta/HfO_2/Pd memristors that were integrated on the chip.

Analogue signal processing and image compression

We first configured the array to implement the discrete cosine transformation (DCT) as a typical example of a linear transformation. The DCT is a Fourier-related transform widely used in digital signal processing and image/video compression and processing[13,46]. Mathematically it can be expressed as

$$y(k)=w(k)\sum_{n=1}^{N}x(n)\cos\left[\frac{\pi}{2N}(2n-1)(k-1)\right],k=1,2,\ldots,N \quad (1)$$

where $w(k)=\begin{cases}1/\sqrt{N}, & k=1,\\ \sqrt{2/N}, & 2\le k\le N\end{cases}$

The equation can also be written as a matrix operation:

$$\mathbf{y}=\mathbf{x}M_{\mathrm{dct}} \quad (2)$$

where $\mathbf{x}$ is the input signals vector, M_{dct} is the DCT matrix, and $\mathbf{y}$ is the output spectrum vector.

One challenge in implementing the DCT with a crossbar is that a memristor conductance value cannot be negative, whereas some of the elements in M_{dct} have negative values. To address this issue, the first approach used here is to map the matrix values into conductance by the linear transformation

$$G_{\mathrm{dct}}=\beta\,M_{\mathrm{dct}}+m_{\mathrm{s}}\,J \quad (3)$$

where J is the matrix of ones, and the transformation coefficients are determined by

$$\beta=[G_{\max}-G_{\min}]/[\max(M_{\mathrm{dct}})-\min(M_{\mathrm{dct}})],\quad m_{\mathrm{s}}=G_{\min}-\beta\min(M_{\mathrm{dct}}) \quad (4)$$

The DCT result can be recovered from the measured output of the crossbar by

$$\mathbf{y}=(\alpha\beta)^{-1}\,\mathbf{i}_{\mathrm{out}}-m_{\mathrm{s}}\beta^{-1}\sum x_i\,\mathbf{j} \quad (5)$$

where $\alpha=\mathbf{v}_{\mathrm{in}}/\mathbf{x}$ is the scaling factor to match the voltage range of the input, $\mathbf{i}_{\mathrm{out}}$ is the vector of output currents, and $\mathbf{j}$ is the vector of ones. The second term of the equation includes a summation over all elements of the input voltages, which can be post-processed by either software or hardware.

The second approach we employed was to use the conductance difference of two memristors (a differential pair) to represent one matrix element. The input voltage signals on two neighbouring rows have the same amplitude, but opposite polarity. The differential calculation is performed by direct current summation:

$$I_{\mathrm{out},j}=\sum_i\left[V_iG_{i,j}^{+}+(-V_i)G_{i,j}^{-}\right]=\sum_i V_i\left(G_{i,j}^{+}-G_{i,j}^{-}\right) \quad (6)$$

where $G_{i,j}^{+}-G_{i,j}^{-}$ is the mapped matrix element in the ith row and jth column and thus can be negative. The differential pair can also mitigate stuck or sticky device issues by setting the conductance of one device in the pair while keeping the other device untouched. This approach provides a level of defect tolerance to the calculation, but at the expense of increasing the number of required memristor cells and thus the chip area.

After configuring one 64×64 crossbar with the aforementioned first approach, the linear transform (equation (3)) to map DCT matrix values to memristor crossbar conductance, we quantitatively analysed the output accuracy of the memristor DCTs by plotting the experimental measurements versus the expected currents for each

column for a range of inputs. The readout conductance matrix after programming into the crossbar array is shown in Supplementary Fig. 4b. The raw current is processed in software by a simple scaling (for details see Supplementary Note 2 and Supplementary Fig. 9), which can be accommodated in hardware with a simple modified design, as the raw column current itself is converted from a voltage output by a TIA. The crossbar output shows an excellent match between the experimental and expected outputs (Fig. 2a). The high accuracy of the DCT reported here mainly resulted from the high bit yield, the relatively low series resistances (0.35 Ω per block for rows, 0.32 Ω per block for columns) and the high I–V linearity of the memristors in the crossbar obtained from the back-end process, as summarized in Supplementary Table 1. The unresponsive devices, especially those stuck in high conductance, have a significantly adverse effect on the output accuracy as well as the power consumption, based on our simulation results shown in Supplementary Fig. 10 and Supplementary Note 3. In the simulation result shown in Supplementary Fig. 11, it is observed that a larger wire series resistance significantly decreases the output accuracy, especially for larger arrays, and eventually impacts the ability to correct the results with a simple linear correction.

Figure 2b shows a typical histogram of the estimated output error for a 64×64 memristor crossbar from all the columns, with input vectors representing image pixel intensities multiplied by a fixed DCT matrix (4,096 data points). The results show that the relative output error nearly follows a normal distribution. Similar analyses were performed with different crossbar sizes and the equivalent bit precision was then extracted from the standard deviations of the estimated output errors. The resulting 5–8 bit precision as a function of crossbar size is shown in Fig. 2c, with larger arrays being systematically less precise. The degradation of bit precision with larger crossbar size could be due to increasing worst-case series wire resistance, leading to significant voltage drops within the array and the presence of increasing sneak currents that cause the conductance states of memristors to influence each other. This can be remedied by decreasing the wire resistances and/or using lower average device state conductance, at the risk of increasing the device nonlinearity. Additionally, using the defect-tolerant approach of differential pairs of devices, described above, also reduces errors.

We start with a one-dimensional (1D) DCT for the crossbar array, to be used as a spectrum analyser, where we employ the conductance matrix used in the above accuracy analysis (the experimentally written values shown in Supplementary Fig. 4b). The input signals are sine waves with different frequencies, the mean values (d.c. components) of which are zero; as a result, they do not require the summation post-processing described above. The experimental crossbar output displays the frequency spectrum, showing good agreement with the software DCT in MATLAB (Fig. 3). The real-time crossbar output with changing input frequencies is also shown in Supplementary Video 2. The input to the crossbar can be directly connected to the analogue output of a sensor or other edge device to directly provide spectral analysis of a signal without the need to digitize it first.

We used the same system for image compression, performing a two-dimensional (2D) DCT. The input image pixel intensities were converted to voltage signals and then applied to the DCT-programmed crossbar, row by row and then column by column, as described in detail in the Methods and Supplementary Fig. 12. Images with pixel counts larger than the crossbar were divided into sub-images, processed in series, and then tiled together after reconstruction (Fig. 4a). In this case we used differential pairs of memristors in neighbouring rows to represent DCT matrix elements, and thus the 64×64 DCT matrix was experimentally represented by the full 128×64 memristor crossbar (Fig. 4b). The 2D cosine transforms of the input images were experimentally acquired from the crossbar, and the amplitudes of the spectra at lower frequencies were much higher than at high frequencies (a typical spectrum is shown in Supplementary Fig. 12f), demonstrating the high energy-compaction and noise-filtering capability of the DCT. We retained the frequencies containing the top 15% of the spectral amplitudes (that is, compression ratio of 20:3) and reconstructed the image using the 2D inverse DCT function in MATLAB to represent data analysis in the Cloud. The results are compared with those using the MATLAB 2D DCT to compress the image in Fig. 4c,d. Different compression ratios ranging from 20:1 to 2:1 were also analysed and compared (Supplementary Fig. 13), showing that even with only 1/20th of the original information we could still reconstruct a reasonable image, even with imperfections, in a memristor crossbar. This demonstration was not optimized for image compression and better results are expected after implementing a quantizer and entropy encoder[47,48].

Convolutional image filtering

We also experimentally demonstrated 2D convolution for image filtering. We used 10 different convolutional filters: Gaussian, disk and average to smooth out noisy images, Laplacian of Gaussian (LoG) with three different parameters, Sobel (both x and y gradient) to extract the edges and Motion (two directions) to mimic the motion blur effect. We added artificial Gaussian white noise

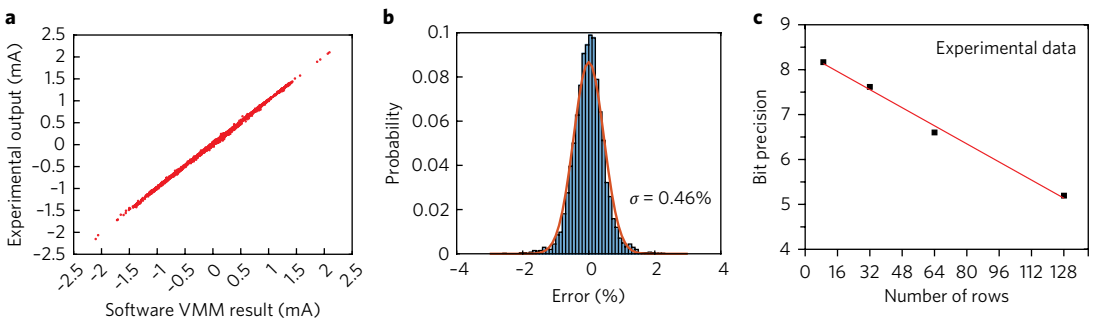

Fig. 2 | Experimental output accuracy and precision for discrete cosine transformation (DCT) using memristor crossbars. a, Relation between experimental output and the software DCT result, showing excellent agreement and thus high accuracy. **b**, Histogram of the DCT estimated output error for the 64×64 crossbar, with a fitted standard deviation of $\sigma = 0.46\%$. The estimated percent error is defined as the difference between the corrected experimental output and the expected value divided by the sensing range. The peak of the corrected error distribution is at 0%. **c**, Bit precision estimated from standard deviations of the output error for crossbars of different sizes. In the bit precision estimate, one discrete output level was defined by $\pm\sigma$. The red line is a linear fit of the data.

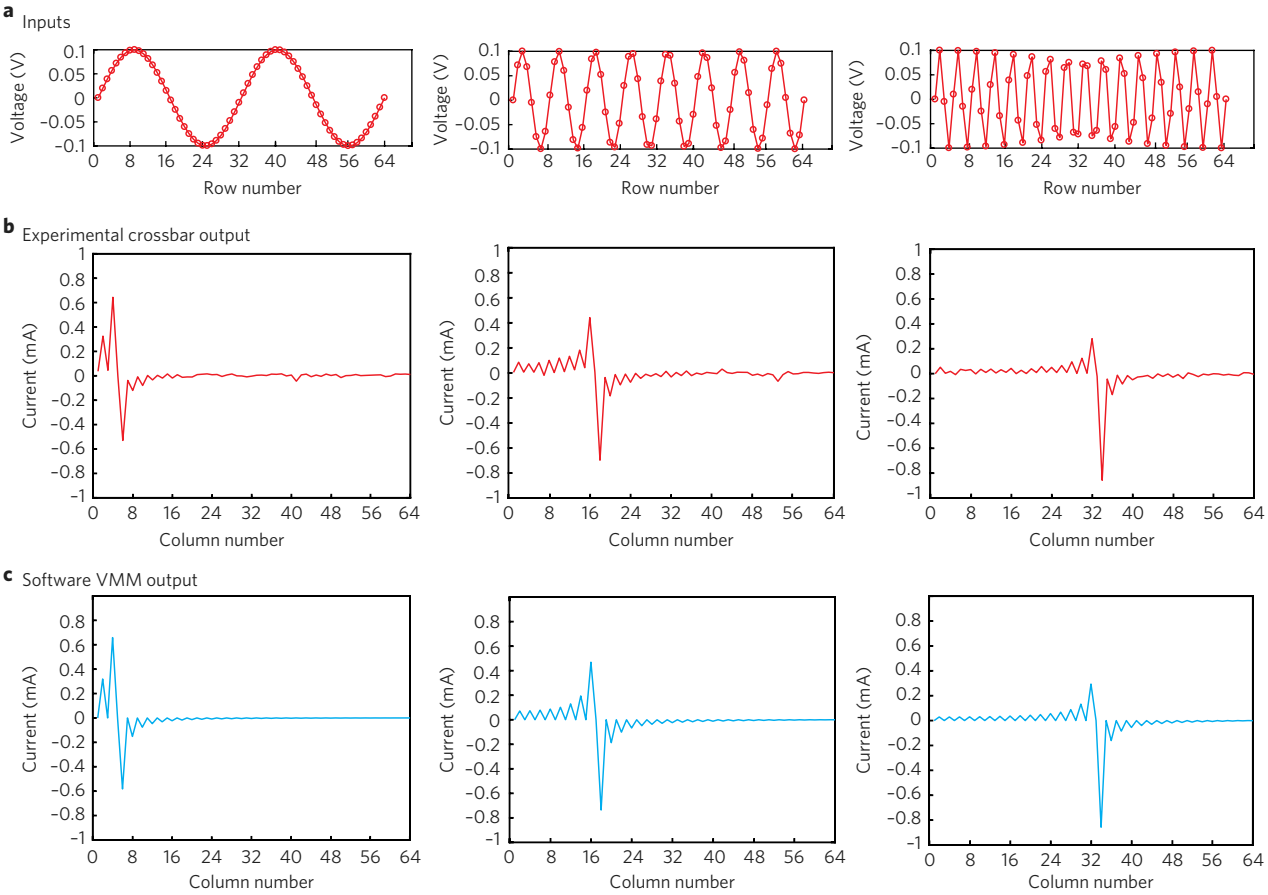

Fig. 3 | Experimental realization of a memristor crossbar-based spectrum analyser. a, Input voltage signals (sine waves) with different frequencies. **b**,**c**, Experimental corrected output (**b**) and MATLAB calculated output (**c**) from the array, showing good agreement. Frequencies are represented by column numbers in this demonstration.

to the original 128×128 Lena image to show how the convolutions damp out noise and are able to locate edges. The noisy Lena image was used as input, the image intensity of which was converted into voltages applied to the rows of the crossbar, as illustrated in Fig. 5a,b. Each pixel in the filtered image was generated by the dot product of the 25-dimensional voltage vector mapped from a 5×5 input sub-image and the 25-dimensional conductance vector mapped from a 5×5 convolution matrix (Supplementary Fig. 13a). We scanned the 5×5 sub-image with a stride of one and did not use zero-padding, so the dimension of the filtered images was 124×124 (=128−5+1). The negative values of the convolution matrices were mapped to memristor cell conductance by the differential approach described earlier, but the differential pairs were arranged in neighbouring columns rather than rows (Fig. 5b). Thus, the 10 different convolution maps were generated in parallel from 20 columns of current output. The experimental results are presented in Fig. 5c, which shows the performance of the crossbar in smoothing images and extracting the edges out of the images, and in Supplementary Fig. 14b for the simple post-processed edges. More results on the original Lena image without noise are shown in Supplementary Fig. 15. The edge extractions described in this step are also a frequent layer of convolutional neural networks (CNNs or ConvNets)[18,49,50], which is the most computationally expensive step in the networks. Compared to previously reported convolutions operating with binary inputs, binary weights and series readout[18], our image filtering procedure included both analogue convolution matrices and analogue inputs, as well as parallel readout of 10 feature maps.

The key advantages of our hardware VMM approach are reconfigurability of the memristor crossbar, reasonable accuracy and precision of the physical computation and efficiency both in speed and energy consumption. Here we analyse the performance and energy efficiency of the system. Because physical multiplication of a 128-dimensional vector and a 128×64 matrix is accomplished by a single current read process on the column wires, a readout time within 10 ns gives 1.64 tera-operations per second (TOPS) (for a detailed discussion see Supplementary Note 4 and Supplementary Fig. 16). We performed a simulation of the power consumption for the image compression task with our experimental parameters, including conductance measurements after programming, dissipation by the wire resistances and writing the input patterns, and found the power consumed in the 128×64 crossbar array was ~13.7 mW, or an efficiency of ~119.7 effective tera-operations per watt. As an approximate comparison, a highly optimized digital system with an application-specific integrated circuit (ASIC) fabricated at the 40 nm technology node for 4-bit 100-dimensional vector and 4-bit 100×200 matrix multiplication, for which the accuracy is comparable with our solution, has a reported energy efficiency of 7.02×10^{12} operations per second per watt[29]. Although not a direct comparison, our system is 17 times more energy-efficient than the ASIC solution. The energy efficiency could be further improved by using memristors that work in a high resistance range but with linear I–V and stable multilevel states, smaller voltage inputs and/or shorter pulses.

A low latency is highly desired for IoT applications such as signal and image processing. The latency of the VMM performed in our

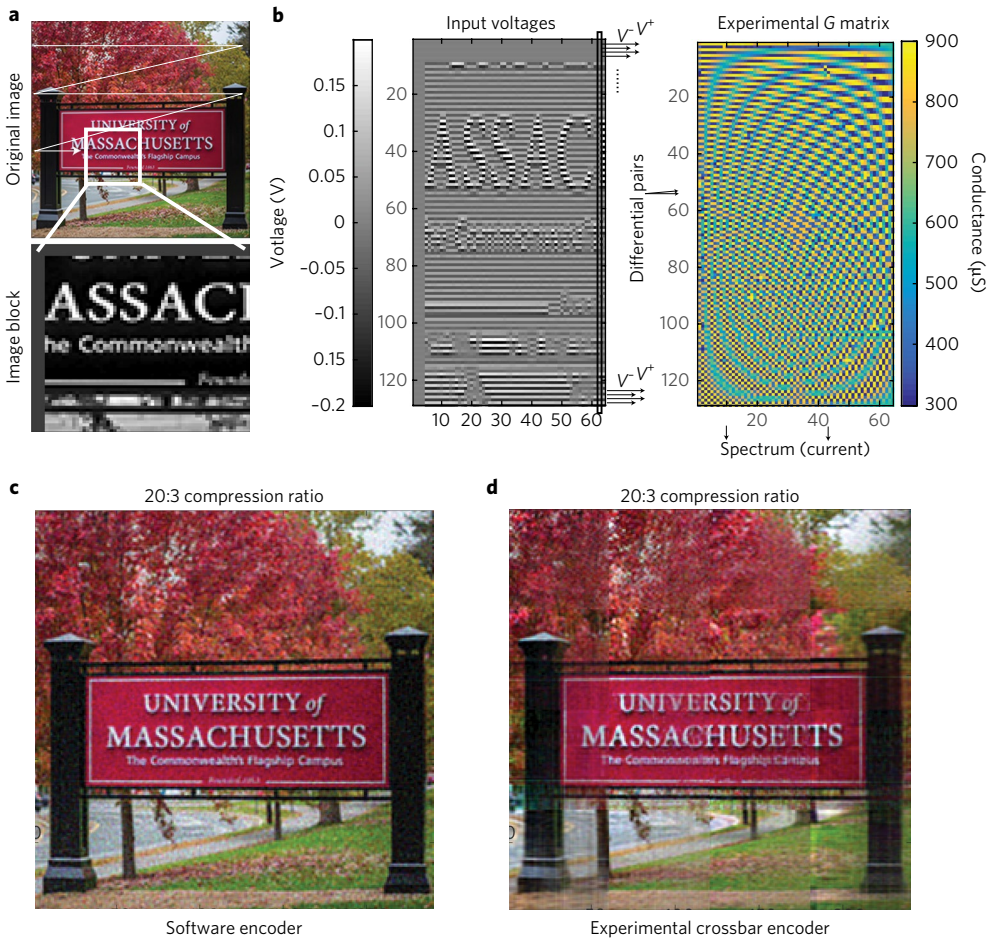

Fig. 4 | Experimental 2D DCT demonstration using differential conductance pairs for image compression and processing. a, The original image for compression was input into the crossbar for the 2D DCT, block by block. The white arrow shows the block processing sequence. The lower image shows a representative image block to be processed. **b**, The image block was converted to voltages that were applied to the row wires of the crossbar (left), with neighbouring wires having a voltage pair with the same amplitude, representing image pixel intensity, but opposite polarity. The paired voltages (represented by the horizontal arrows) were applied to a differential pair of memristor conductance, with the resulting net current representing the product of the absolute voltage value and the conductance difference of the differential pair. Right: differential DCT written into the 128 × 64 array, with the small number of stuck 'on' or 'off' memristors evident as disruptions in the pattern. **c**,**d**, Images decoded from the 2D DCT by software (**c**) and experimentally (**d**). Before decoding, only the frequencies representing the top 15% of the spectral intensity were preserved (a 20:3 compression ratio).

memristor array is a one-step current readout on the column wires, which does not scale up with increased input vector dimension. This is advantageous over a digital system whose latency inevitably increases with the input dimension, because the multiplication and summation have to be calculated step by step. More importantly, our memristor crossbar hardware VMM can process analogue signals acquired from a sensor directly, without the need for extra peripherals such as analogue-to-digital converters (ADCs), which would be required for a digital ASIC solution and consume extra time and energy, but was not considered in the above energy estimation. Additionally, high-bit precision ADCs after crossbar columns are not necessary if only specific features need to be detected within signals, which can be provided with threshold-gate circuits at much lower cost both in latency and energy. This flexibility, along with low latency and high energy efficiency, make analogue crossbar computation ideal for a wide range of edge and IoT computations.

Conclusions

We have demonstrated analogue-vector and analogue-matrix-vector multiplication using crossbars with over 8,000 memristors, with an equivalent 6-bit or 64-level precision and 99.8% device yield. The device conductance states were precisely tuned and the *I*–*V* characteristics were linear, ideal for analogue computing. We have successfully implemented some important applications for IoT and edge computing, including signal processing, image compression and convolutional filtering. The energy efficiency of the system was over 119.7 trillion equivalent operations per second per watt using a readout of 10 ns, and this is expected to increase significantly with larger vectors and matrices and with improvements in circuitry. Our results are an encouraging advance in the hardware implementation of computing using emerging devices, and provide a promising path towards energy-efficient analogue computing based on memristors.

Methods

Memristor fabrication and integration. The transistors arrays were fabricated in a commercial laboratory with minimized wire resistance. For demonstration purpose with reduced cost, the transistors had a feature size of 2 μm and the fabrication did not involve a planarization process. The memristor arrays were fabricated in house using photolithography, thin-film deposition and liftoff. Specifically, argon plasma treatment was performed on the as-received CMOS chip to remove native metal oxide layers for better electrical connection, followed by the sputtering of 5 nm Ag and 200 nm Pd as metal vias. After lifting off in warm acetone, the sample was annealed at 300 °C for half an hour in 20 s.c.c.m. nitrogen flow. A 60-nm-thick

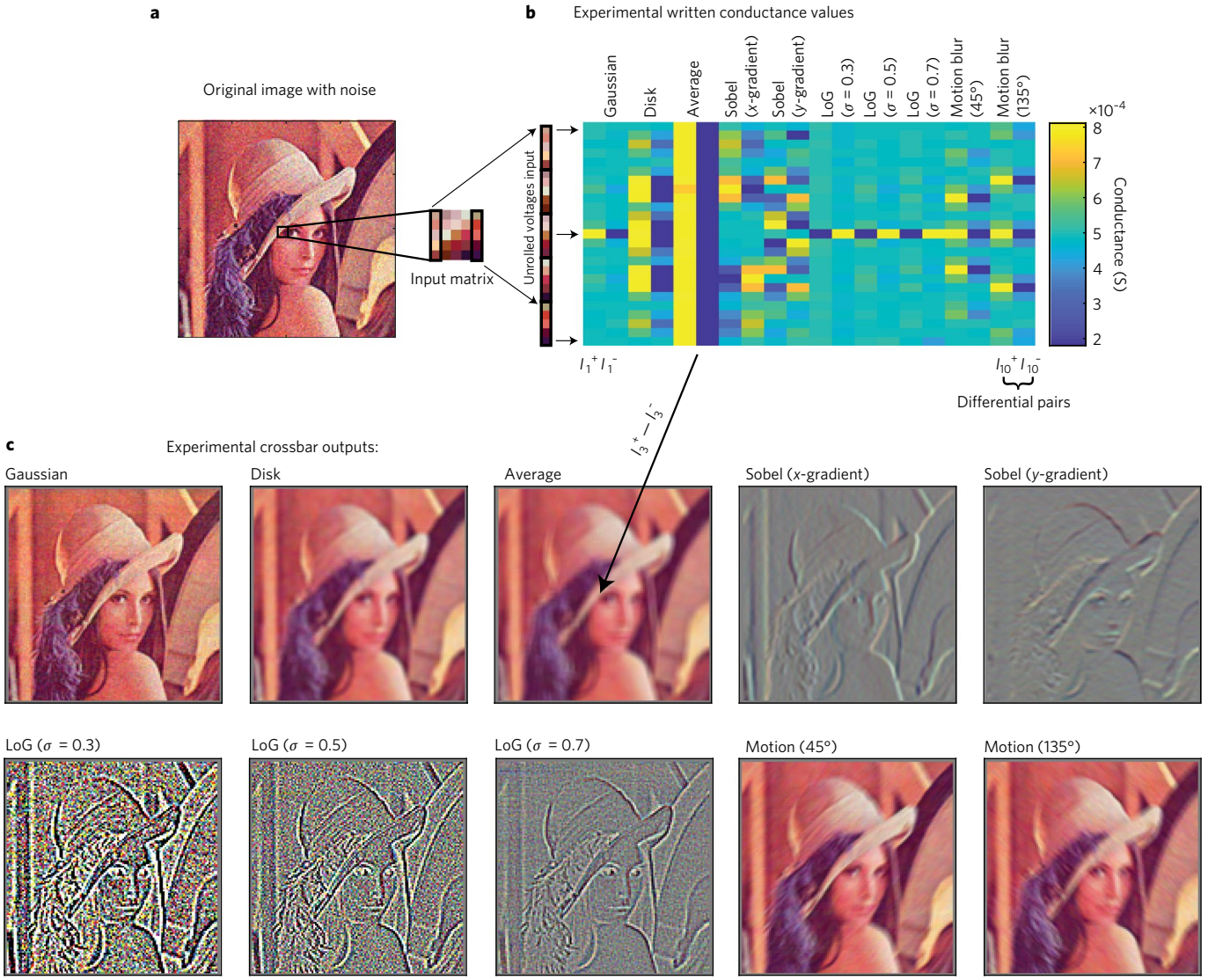

Fig. 5 | Experimental convolution demonstration with differential memristor conductance pairs. a, The input image was a standard Lena image with artificially added Gaussian white noise. Each colour channel of the image is represented by a floating-point number between 0 and 1, and the added noise has a standard deviation of 0.004 and zero mean. **b**, Measured conductance after programming 10 convolutional filters into the 25×20 crossbar. The pixel intensities, represented by two voltages with equal amplitude and opposite polarity, were input into the crossbar onto a pair of memristors in adjacent columns. The difference in the conductance of the memristor pair represents one matrix element of a convolution. **c**, Ten different filtered images obtained in parallel by the convolution operation: Gaussian, disk and average reduce noise by smoothing the image, Laplacian of Gaussian (LoG) with various parameters and Sobel (*x* and *y* gradient) were used to detect edges, and Motion to generate motion blur.

Pd/5 nm Ta adhesive layer was then sputtered as the bottom electrode. The 5 nm HfO_2 switching layer was deposited by atomic layer deposition (ALD) using water and tetrakis(dimethylamido)hafnium as precursors at 250 °C. Patterning of the switching layer was carried out by photolithography and reactive ion etch (RIE) using CHF_3/O_2 chemistry. Finally, a 50-nm-thick Ta layer was sputtered and lifted off to serve as the top electrode, covered with another 10-nm-thick Pd layer as the passivation layer.

Electrical characterization. Most electrical characterization was carried out using our custom-built multiboard measurement system[51]. A photograph and description of the system are provided in Supplementary Fig. 2.

2D DCT steps for image compression. The 2D spectra of an image could be acquired in two steps of matrix multiplication. In the first step of the 2D DCT of an image, every row of the image intensities was converted to a voltage amplitude vector and applied to the row wires of the crossbar (Supplementary Fig. 12a). It is noteworthy that the voltage amplitude may come from the direct analogue output of an image sensor. In this case, the conductances of the 128×64 memristor array were mapped from a 64×64 DCT matrix with a row differential method (Supplementary Fig. 12b). The image intensities of each row, with 64 pixels, were converted following the differential requirement into a 128-dimensional voltage vector. Specifically, neighbouring voltage vector elements have the same amplitude representing one image pixel intensity, but with different polarity. As a result, the current outputs on the columns of the crossbar array are naturally the VMM result of the input voltages vector and conductance matrix. The output current matrix, in which each row is one VMM result with one row of image intensity as input, is shown in Supplementary Fig. 12c. Each row of the output matrix is the current vector output when applying one row of voltage vectors and is thus the spectrum of the input image along the horizontal direction after the cosine transform. The second step DCT calculates the spectrum along the vertical direction, so the output matrix from the first step is transposed and linearly mapped into the voltage input matrix for the second step DCT (Supplementary Fig. 12d). The voltages are then applied on the rows of the crossbar, similarly to the first step, without changing the conductance matrix in the crossbar (Supplementary Fig. 12e). The output current matrix in this step (shown in Supplementary Fig. 12f) is the 2D DCT result that represents the 2D spectra of the input image.

Data availability. The data that support the plots within this paper and other findings of this study are available from the corresponding author upon reasonable request.

Received: 2 June 2017; Accepted: 19 October 2017;
Published online: 4 December 2017

References

1. Williams, R. S. What's next? *Comput. Sci. Eng.* **19**, 7–13 (2017).
2. Waldrop, M. M. The chips are down for Moore's law. *Nature* **530**, 144–147 (2016).
3. Gubbi, J., Buyya, R., Marusic, S. & Palaniswami, M. Internet of Things (IoT): a vision, architectural elements, and future directions. *Fut. Gen. Comput. Syst.* **29**, 1645–1660 (2013).
4. Yocam, E. W. Evolution on the network edge: intelligent devices. *IT Professional* **5**, 32–36 (2003).
5. Chua, L. Memristor—the missing circuit element. *IEEE Trans. Circuit Theory* **18**, 507–519 (1971).
6. Strukov, D. B., Snider, G. S., Stewart, D. R. & Williams, R. S. The missing memristor found. *Nature* **453**, 80–83 (2008).
7. Yang, J. J., Strukov, D. B. & Stewart, D. R. Memristive devices for computing. *Nat. Nanotech.* **8**, 13–24 (2013).
8. De Salvo, B. *Silicon Non-Volatile Memories: Paths of Innovation* (Oxford, Wiley, 2013).
9. Wong, H.-S. P. et al. Metal–oxide RRAM. *Proc. IEEE* **100**, 1951–1970 (2012).
10. Ventra, M. D., Pershin, Y. V. & Chua, L. O. Circuit elements with memory: memristors, memcapacitors, and meminductors. *Proc. IEEE* **97**, 1717–1724 (2009).
11. Truong, S. N. & Min, K.-S. New memristor-based crossbar array architecture with 50% area reduction and 48% power saving for matrix-vector multiplication of analog neuromorphic computing. *J. Semicond. Technol. Sci.* **14**, 356–363 (2014).
12. Xia, L. et al. Technological exploration of RRAM crossbar array for matrix-vector multiplication. *J. Comput. Sci. Technol.* **31**, 3–19 (2016).
13. Li, B., Gu, P., Wang, Y. & Yang, H. Exploring the precision limitation for RRAM-based analog approximate computing. *IEEE Design Test* **33**, 51–58 (2016).
14. Prezioso, M. et al. Training and operation of an integrated neuromorphic network based on metal–oxide memristors. *Nature* **521**, 61–64 (2015).
15. Yu, S. et al. in *Proc. Int. Electron Dev. Meet.* 416–419 (San Francisco, IEEE, 2016).
16. Park, S. et al. Electronic system with memristive synapses for pattern recognition. *Sci. Rep.* **5**, 10123 (2015).
17. Hu, M. & Strachan, J. P. in *Proc. 2016 IEEE Int. Conf. Rebooting Comp. (ICRC)* 1–5 (San Diego, IEEE, 2016).
18. Gao, L., Chen, P.-Y. & Yu, S. Demonstration of convolution kernel operation on resistive cross-point array. *IEEE Electron Dev. Lett.* **37**, 870–873 (2016).
19. Indiveri, G., Linares-Barranco, B., Legenstein, R., Deligeorgis, G. & Prodromakis, T. Integration of nanoscale memristor synapses in neuromorphic computing architectures. *Nanotechnology* **24**, 384010 (2013).
20. Park, J. et al. TiO_x-based RRAM synapse with 64-levels of conductance and symmetric conductance change by adopting a hybrid pulse scheme for neuromorphic computing. *IEEE Electron Dev. Lett.* **37**, 1559–1562 (2016).
21. Fumarola, A. et al. in *Proc. 2016 IEEE Int. Conf. Rebooting Comp. (ICRC)* 1–8 (San Diego, IEEE, 2016).
22. Ge, N. et al. An efficient analog Hamming distance comparator realized with a unipolar memristor array: a showcase of physical computing. *Sci. Rep.* **7**, 40135 (2017).
23. Hu, M. et al. in *Proc. 53rd Design Automat. Conf.* 1–6 (Austin, ACM, 2016).
24. Gao, L., Alibart, F. & Strukov, D. B. in *IEEE/IFIP 20th Int. Conf. VLSI and System-on-Chip, 2012 (VLSI-SoC)* 88–93 (Santa Cruz, IEEE, 2012).
25. Chakrabarti, B. et al. A multiply-add engine with monolithically integrated 3D memristor crossbar/CMOS hybrid circuit. *Sci. Rep.* **7**, 42429 (2017).
26. Lastras-Montaño, M. A., Chakrabarti, B., Strukov, D. B. & Cheng, K. T. in *Design, Automation & Test in Europe Conference & Exhibition (2017)* 1257–1260 (Lausanne, IEEE, 2017).
27. Ma, W. et al. in *Proc. Int. Electron Dev. Meet.* 436–439 (San Francisco, IEEE, 2016).
28. Yao, P. et al. Face classification using electronic synapses. *Nat. Commun.* **8**, 15199 (2017).
29. Sheridan, P. M. et al. Sparse coding with memristor networks. *Nat. Nanotech.* **12**, 784–789 (2017).
30. Choi, S., Shin, J. H., Lee, J., Sheridan, P. & Lu, W. D. Experimental demonstration of feature extraction and dimensionality reduction using memristor networks. *Nano Lett.* **17**, 3113–3118 (2017).
31. Jouppi, N. P., Young, C., Patil, N. & Patterson, D. in *44th Int. Symp. Comp. Archit. (ISCA)* 1–17 (ACM/IEEE, Toronto, 2017).
32. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
33. Dally, W. in *Neural Information Processing Systems (NIPS2015) Tutorial* (NIPS Foundation, Montréal, 2015).
34. Shafiee, A. et al. in *2016 ACM/IEEE 43rd Int. Symp. Comp. Archit. (ISCA)* 14–26 (Seoul, IEEE, 2016).
35. Hu, M., Li, H., Wu, Q. & Rose, G. S. in *2012 49th ACM/EDAC/IEEE Design Automat. Conf. (DAC)* 498–503 (San Francisco, IEEE, 2012).
36. Jiang, H. et al. Sub-10 nm Ta channel responsible for superior performance of a HfO_2 memristor. *Sci. Rep.* **6**, 28525 (2016).
37. Linn, E., Rosezin, R., Kugeler, C. & Waser, R. Complementary resistive switches for passive nanocrossbar memories. *Nat. Mater.* **9**, 403–406 (2010).
38. Kim, K. M. et al. Low-power, self-rectifying, and forming-free memristor with an asymmetric programming voltage for a high-density crossbar application. *Nano Lett.* **16**, 6724–6732 (2016).
39. Li, C. et al. Three-dimensional crossbar arrays of self-rectifying $Si/SiO_2/Si$ memristors. *Nat. Commun.* **8**, 15666 (2017).
40. Midya, R. et al. Anatomy of Ag/Hafnia-based selectors with 10^{10} nonlinearity. *Adv. Mater.* **29**, 1604457 (2017).
41. Jo, S. H., Kumar, T., Narayanan, S. & Nazarian, H. Cross-point resistive RAM based on field-assisted superlinear threshold selector. *IEEE Trans. Electron Dev.* **62**, 3477–3481 (2015).
42. Choi, B. J. et al. Trilayer tunnel selectors for memristor memory cells. *Adv. Mater.* **28**, 356–362 (2016).
43. Ji, L. et al. Integrated one diode–one resistor architecture in nanopillar SiO_x resistive switching memory by nanosphere lithography. *Nano Lett.* **14**, 813–818 (2014).
44. Van Wees, B. J. et al. Quantized conductance of point contacts in a two-dimensional electron gas. *Phys. Rev. Lett.* **60**, 848–850 (1988).
45. Yi, W. et al. Quantized conductance coincides with state instability and excess noise in tantalum oxide memristors. *Nat. Commun.* **7**, 11142 (2016).
46. Rao, K. R. & Yip, P. *Discrete Cosine Transform: Algorithms, Advantages, Applications* (Cambridge, Academic Press Professional, 1990).
47. Pennebaker, W. B. & Mitchell, J. L. *JPEG: Still Image Data Compression Standard* (Berlin, Springer Science & Business Media, 1992).
48. Malarvizhi, D. & Kuppusamy, D. K. A new entropy encoding algorithm for image compression using DCT. *Int. J. Eng. Trends Technol.* **3**, 327–332 (2012).
49. Krizhevsky, A., Sutskever, I. & Hinton, G. E. in *Advances in Neural Information Processing Systems 25 (NIPS 2012)* 1097–1105 (Stateline, NV, NIPS Foundation, 2012).
50. Lawrence, S., Giles, C. L., Ah Chung, T. & Back, A. D. Face recognition: a convolutional neural-network approach. *IEEE Trans. Neural Networks* **8**, 98–113 (1997).
51. Hu, M. et al. Memristor-based analog computation and neural network classification with a dot product engine. *Adv. Mater.* https://doi.org/10.1002/adma.201705914 (in the press).

Acknowledgements

This work was supported in part by the Air Force Research Laboratory (AFRL; grant no. FA8750-15-2-0044), the US Air Force Office for Scientific Research (AFOSR; grant no. FA9550-12-1-0038), the Intelligence Advanced Research Projects Activity (IARPA; contract 2014-14080800008) and the National Science Foundation (NSF; ECCS-1253073). This work was performed in part at the Center for Hierarchical Manufacturing (CHM), an NSF sponsored Nanoscale Science and Engineering Center (NSEC) at University of Massachusetts, Amherst.

Author contributions

C.L., H.J., N.G., N.D., P.L. and Z.W. built the integrated chips. C.L., M.H., Y.L. and J.P.S. carried out the measurements. E.M., M.H. and J.P.S. built the measurement system. Y.L., M.H. and W.S. performed circuit simulation. J.Z. took the cross-sectional SEM and TEM images. J.P.S., J.J.Y. and Q.X. designed the experiments and supervised the project. Q.X., C.L., J.J.Y. and R.S.W. wrote the manuscript. All authors contributed to analysis of the results and commented on the manuscript.

Competing interests

The authors declare no competing financial interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41928-017-0002-z.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence and requests for materials should be addressed to J.P.S. or J.J.Y. or Q.X.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.