nature electronics

SUPPLEMENTARY INFORMATION

https://doi.org/10.1038/s41928-020-0422-z

In the format provided by the authors and unedited.

Gesture recognition using a bioinspired learning architecture that integrates visual data with somatosensory data from stretchable sensors

Ming Wang[1,3], Zheng Yan[2,3], Ting Wang[1], Pingqiang Cai[1], Siyu Gao[1], Yi Zeng[1], Changjin Wan[1], Hong Wang[1], Liang Pan[1], Jiancan Yu[1], Shaowu Pan[1], Ke He[1], Jie Lu[2] and Xiaodong Chen[1]✉

[1]Innovative Centre for Flexible Devices (iFLEX), Max Planck–NTU Joint Lab for Artificial Senses, School of Materials Science and Engineering, Nanyang Technological University, Singapore, Singapore. [2]Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, New South Wales, Australia. [3]These authors contributed equally: Ming Wang, Zheng Yan. ✉e-mail: chenxd@ntu.edu.sg

Supplementary Information for

Human gesture recognition using a bioinspired learning architecture that integrates visual data with somatosensory data from stretchable sensors

Ming Wang[1,3], Zheng Yan[2,3], Ting Wang[1], Pingqiang Cai[1], Siyu Gao[1], Yi Zeng[1], Changjin Wan[1], Hong Wang[1], Liang Pan[1], Jiancan Yu[1], Shaowu Pan[1], Ke He[1], Jie Lu[2], Xiaodong Chen[1]★

Table of Contents:

[3]These authors contributed equally: Ming Wang, Zheng Yan.

★Corresponding author. E-mail: chenxd@ntu.edu.sg

I: Supplementary Notes

1. Normalization process for strain data

Due to device variation and hysteresis, a normalization step to preprocess the strain resistance values measured by strain sensors is required before implementing machine learning. Five transparent and stretchable strain sensors were patched on one right hand of the volunteer. The tiny enameled wires were used to connect the sensors and the measurement equipment, which are almost invisible in the image of 160 × 120 pixels. The resistance values of strain sensors were then recorded when the volunteers made the specified gestures (from I to X).

Each sensor that was mounted on each finger of each volunteer was individually calibrated. Firstly, determine the bottom baseline ($R_{mid-min}$) of the resistance values of the sensor. Here, $R_{mid-min}$ is the median value of multiple measurements (21 times) when a finger of the volunteer is fully extended. The measured relative resistance change ($\Delta R/R_0$) can be obtained by using the following formula:

$$\Delta R/R_0 = \frac{R - R_{mid-min}}{R_{mid-min}} \times 100\%$$

where R represents the raw measured resistance value of the strain sensor. Supplementary Figure 4b shows the measured $\Delta R/R_0$ of the index finger of one volunteer corresponding to different hand gestures (from I to X).

Secondly, determine the top baseline ($R_{mid-max}$) of the resistance values of the sensor. Here, $R_{mid-max}$ is the median value of multiple measurements (21 times) when a finger of the volunteer is under the maximum bending state (making a fist).

The relative $R_{mid-max}$ is marked as $\Delta R_{mid-max}$. Using $\Delta R_{mid-max}$, we can normalize the measured $\Delta R/R_0$, by using the following formula:

$$\text{Normalized } \Delta R/R_0 = \frac{\Delta R/R_0}{\Delta R_{mid-max}}$$

Supplementary Figure 4c shows the normalized $\Delta R/R_0$ corresponding to the measurement results of Supplementary Fig. 4c.

2. AlexNet CNN as a visual feature extractor

Firstly, the AlexNet CNN was pretrained on over a million images (ILSVRC) to classify images into 1000 object categories. The pretrained AlexNet was then taken as a starting point to learn a new task by performing fine-tuning with transfer learning. We truncated the final classification layer of the network and replaced it a new softmax layer that is relevant to our visual perception problem. The new classification layer is of 10 categories instead of 1000. We also replaced the second fully connected layer that has 4096 neurons with a new one with 48 neurons. The reason is that we expect to take the output from this layer as discriminative features. During the training, we resized each image to 227 × 227 pixels to make it compatible with the image of AlexNet to leverage feature representation learned via the ImageNet dataset. The effort for the pretraining is minimal as there are well-trained models available off the shelf.

The CNN is trained via backpropagation with stochastic gradient descent with momentum using a global momentum of 0.9. The first two convolution layers are frozen during the training as they already have pretty good weights to capture

universal features like curves and edges. The subsequent transferred layers are finetuned using a learning rate of 0.0001 and the new layers are trained using a learning rate of 0.001. We do not perform any data augmentation to ensure minimal fine-tuning efforts.

3. Frobenius condition number-dependent pruning strategy

Building sparse neural networks via pruning the dense neural networks is one of the most important approaches to achieve sparse weight parameters while maintaining performance competitive with dense networks. In previous reports, low-ranking weighting parameters are removed to achieve the sparse neural networks, where the ranking often depends on the absolute values of the weights between neurons[S1]. However, the value of the weight might not be proportional to its significance to the overall architecture. Here, we propose a pruning strategy to build a sparse neural network by computing the variation of condition numbers. To the best of our knowledge, it is the first attempt to achieve a sparse neural network by considering the Frobenius condition number of the corresponding weight matrix.

Motivated by the stability theory of linear systems[S2], we prune these connected weights if their removal does not lead to significant numerical increase of the Frobenius condition number of the corresponding weight matrix. The details are described as follows:

Denote W as the weighting matrix between two adjacent layers l and $l+1$ of a dense neural network,

$$W = \{w_{ij}\}_{i=1\ldots m, j=1\ldots n}^{l,l+1}$$

where w_{ij} is the weight between the neuron i in the layer l and the neuron j in the layer $l+1$, m is the number of neurons in the layer l, and n is the number of neurons in the layer $l+1$. The Frobenius condition number of W can be computed as

$$\sigma_F(W) = \frac{\lambda_{\max}(W^T W)}{\lambda_{\min}(W^T W)}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ represent the largest and smallest eigenvalues of the weighting matrix, respectively, and W^T denotes the transpose of W. In view of this formulation, the pruning algorithm is illustrated as follows:

for i=1:m **do**

for j =1:n **do**

$w_{ij} = 0$ and compute $\sigma_F(\widetilde{W})$ where all elements of $\widetilde{W}$ are equal to W except for w_{ij}.

if $\frac{\sigma_F(\widetilde{W})}{\sigma_F(W)} \leq \varepsilon$, where ε is a user-defined threshold (1.2 is used in our experiments), **then**

Connection between the corresponding neurons is removed.

else,

w_{ij} is reset to the original value.

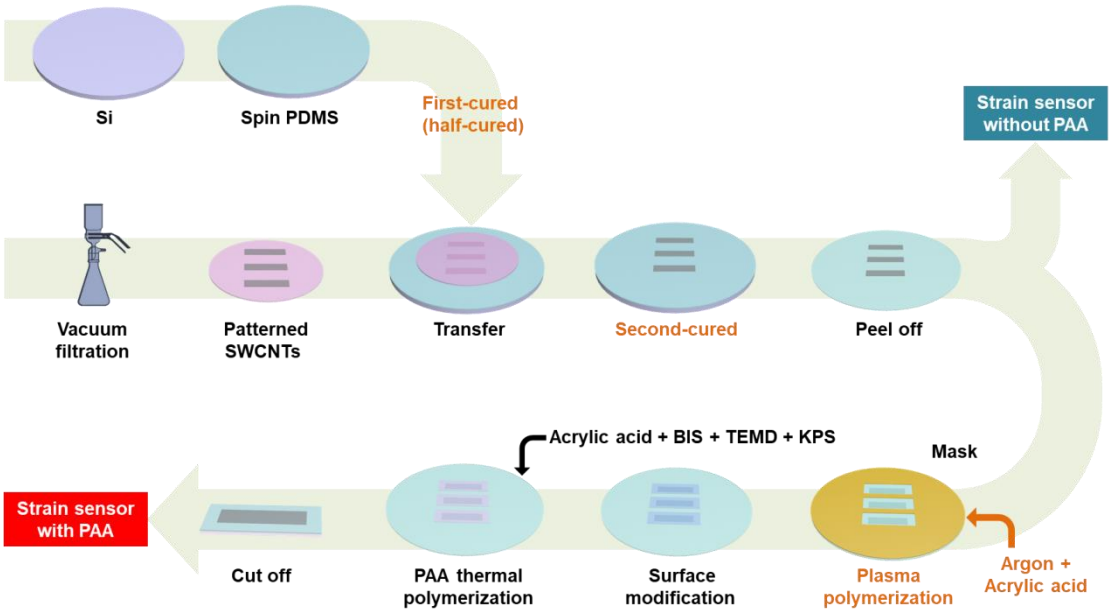
end if

end for

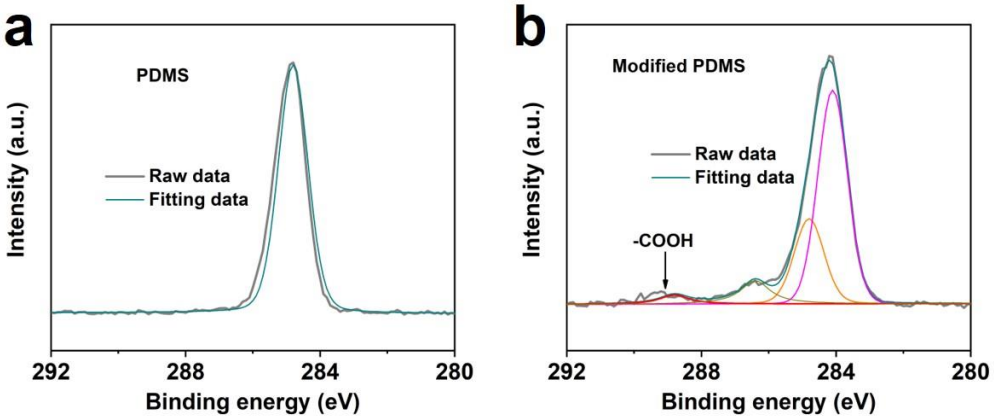
end for

The merits of the above approach are its comprehensive and quantitative evaluation of the effect of every connection (weight) between neurons in the Frobenius condition number-dependent pruning strategy. The resultant sparse model exhibits a high recognition accuracy (100%) using the custom dataset, which can compete with the performance level using a dense neural network (Supplementary Table 1). Furthermore, we evaluate the pruning strategy compared to a common pruning strategy based on the absolute values of the weights (Supplementary Table 1). Under the same network sparsity and experimental settings, the classification accuracy of the BSV architecture based on absolute value-depend pruning strategy is 97.8%, which is 2.2% lower than that of the BSV architecture based on Frobenius condition number-depend pruning strategy. This result demonstrates that the Frobenius condition number-depend pruning strategy is superior to the absolute value-depend one. The reason is that our pruning strategy takes a global correlation into consideration to remove the connections between neurons, which could learn more general weights for making decision.

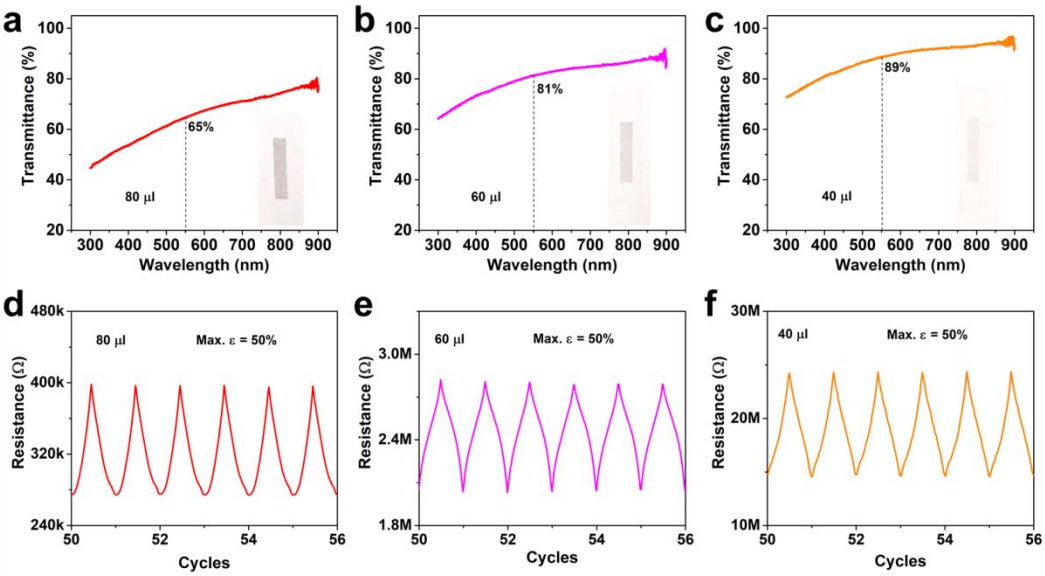
II: Supplementary Figures and Captions



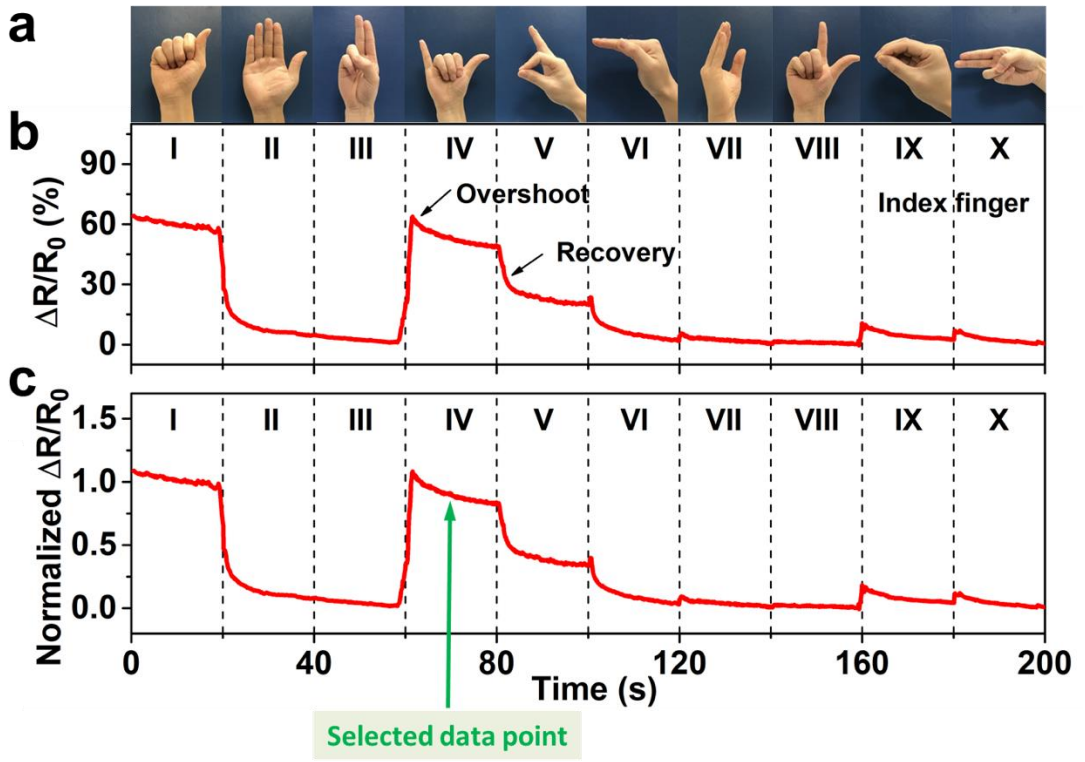
Supplementary Figure 1 Schematic of the fabrication process of strain sensor device. Firstly, the silicone prepolymer and crosslinker mixed with 10: 1 was spin coated on Si wafer and was half-cured (First-cured). Then, the patterned SWCNT film via vacuum filtration method was transferred to the half-cured PDMS substrate. After second-cured, the strain sensor without PAA hydrogels was obtained. In order to add an adhesion layer, PAA hydrogels were *in-situ* polymerized onto the strain sensor. Prior to the PAA polymerization, the plasma polymerization preprocess was implemented to modify the surface of the PDMS substrate (backside to SWCNT film). Then, the solution with water, acrylic acid, BIS, TEMED, KPS in proportion to 1 ml: 0.2 ml: 0.5 mg: 7 μl: 35 mg was dropped onto the PDMS surface. After the PAA polymerization under a hotplate of 70 ˚C for about 20 min, it was cut off into the final device.



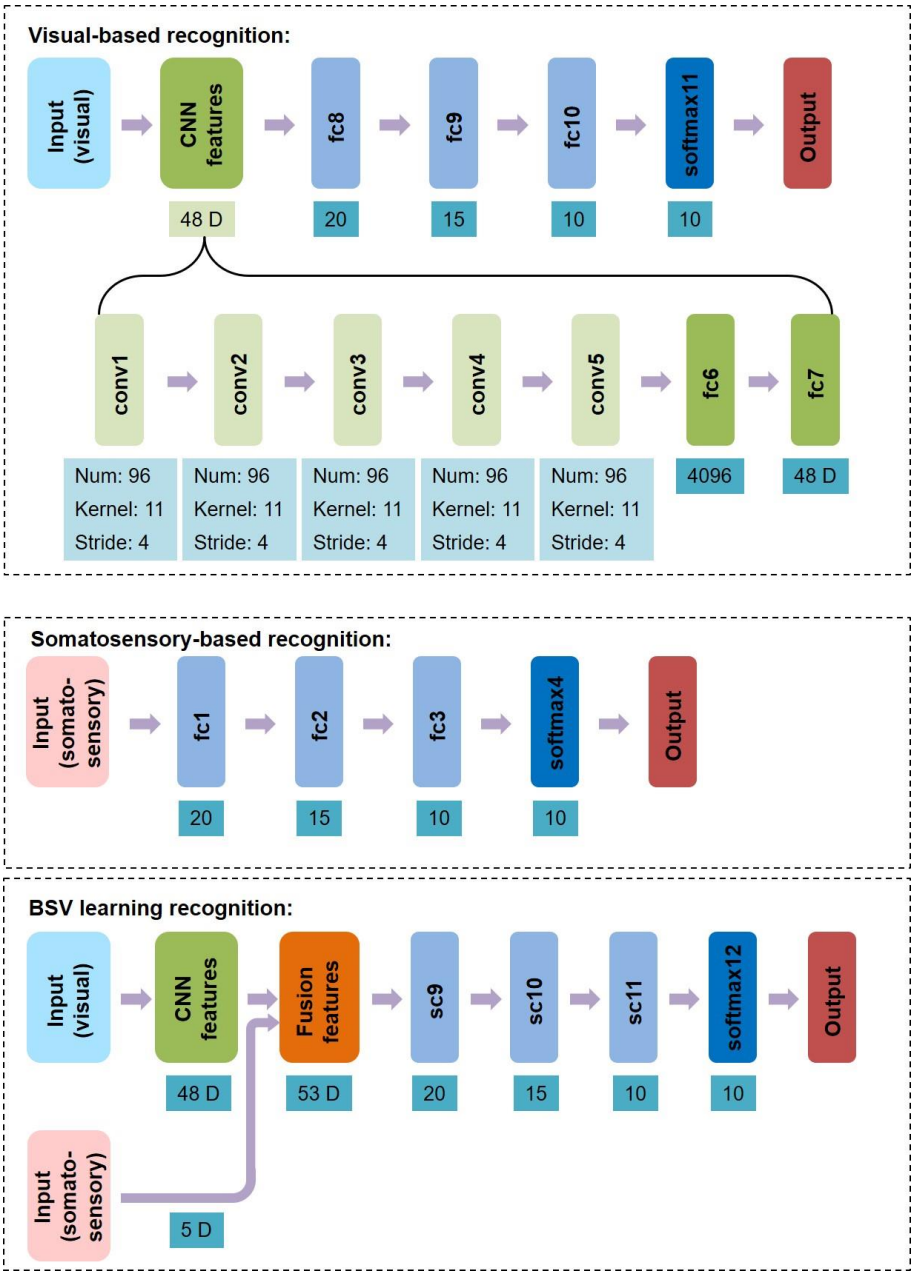
Supplementary Figure 2 XPS spectra of PDMS (a) and PDMS modified by plasma polymerization.



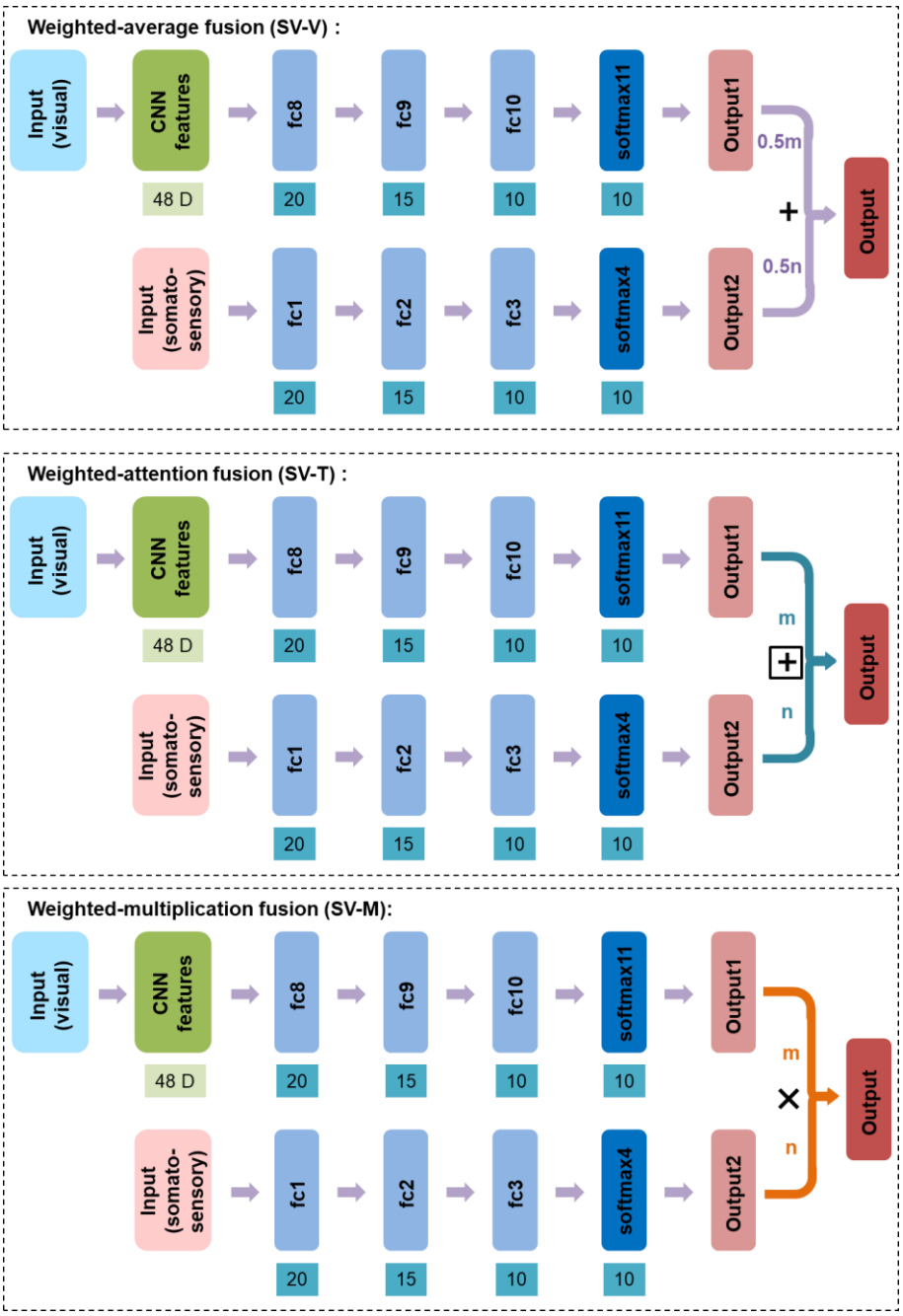
Supplementary Figure 3 Optimization of the strain sensor performance/optical transparency trade-off. (a)-(c), Transmittance of the pure SWCNT layer in the stretchable strain sensors without PAA. The usage of the SWCNT solution for each device was 80, 60, 40 µl. The insets show the photo images of the stretchable strain sensors without PAA. (d)-(f), Strain-resistance response curves corresponding to the devices with different optical transparency. Finally, the stretchable strain sensor with 40 µl SWCNT solution was chosen to collect the somatosensory information to guarantee the reliability in our case.



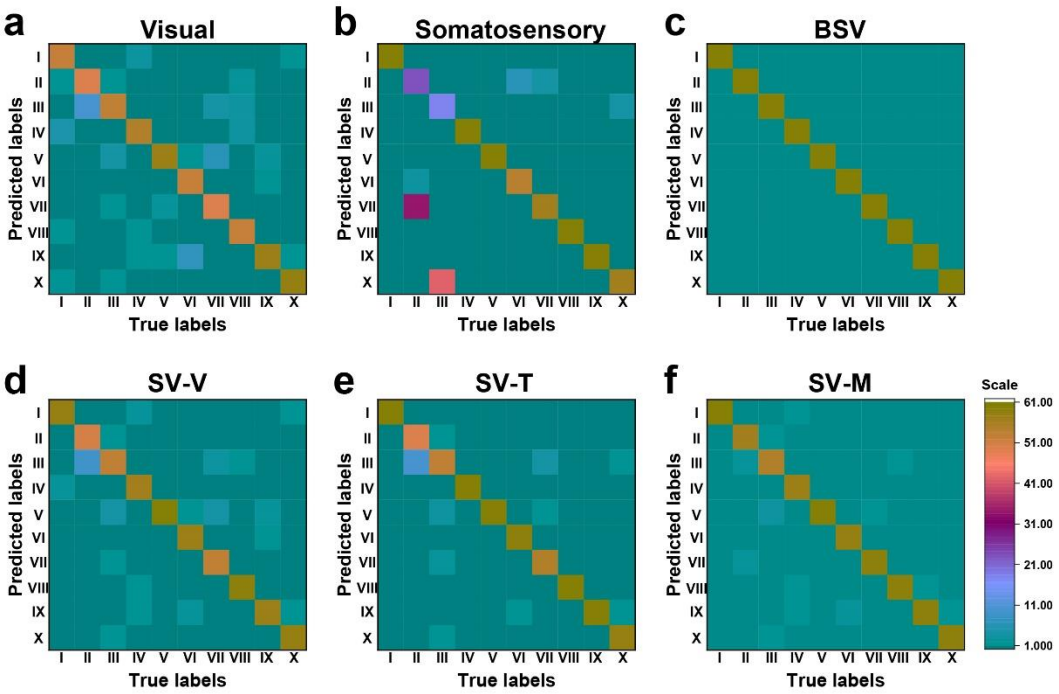
Supplementary Figure 4 Somatosensory information collection of the index finger in one hand gesture. (a) Strain sensor patched over the knuckle of the index finger for ten hand gestures. (b) Relative resistance change ($\Delta R/R_0$) in index finger corresponding to different hand gestures. (c) Normalized $\Delta R/R_0$. The normalized $\Delta R/R_0$ at 10 seconds after each hand gesture changed was chosen as the somatosensory data. For example, the normalized $\Delta R/R_0$ data that green arrow pointed represents the somatosensory information of the index finger in the hand gesture of IV.



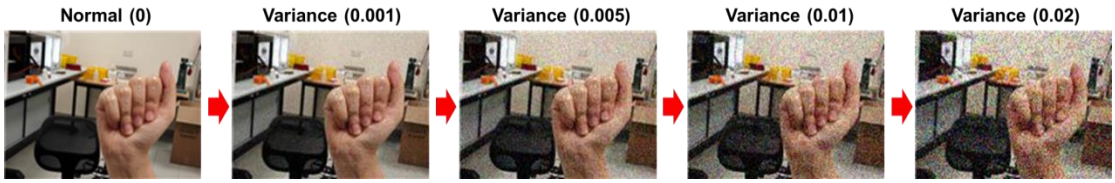
Supplementary Figure 5 Architectures of three recognition strategies. Unimodal strategies: visual-based recognition only using visual images, and somatosensory-based recognition only using strain sensor data. BSV associated learning recognition strategy using both visual images and strain sensor data. Alex CNN for visual-based recognition, a 5-layer feedforward neural network for somatosensory-based recognition. fc: fully connected neural network; sc: sparsely connected neural network.



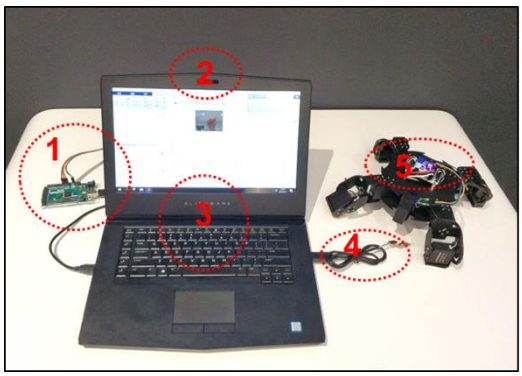
Supplementary Figure 6 Architectures of three common SV fusion recognition strategies: weighted-average fusion (SV-V), weighted-attention fusion (SV-T), and weighted-multiplication fusion (SV-M) using both visual images and strain sensor data.



Supplementary Figure 7 Confusion matrices comparison between six recognition strategies: Visual, somatosensory, BSV, SV-V, SV-T, and SV-M.



Supplementary Figure 8 Processing the normal testing dataset with Gaussian white noise. MATLAB function: imnoise(testing_image, 'gaussian', 0, var_gauss). The testing images (visual information) were processed by adding the Gaussian white noise with mean zero and variance var_gauss (0.001, 0.005, 0.01, and 0.02).



1: DAQ ——Somatosensory information ;

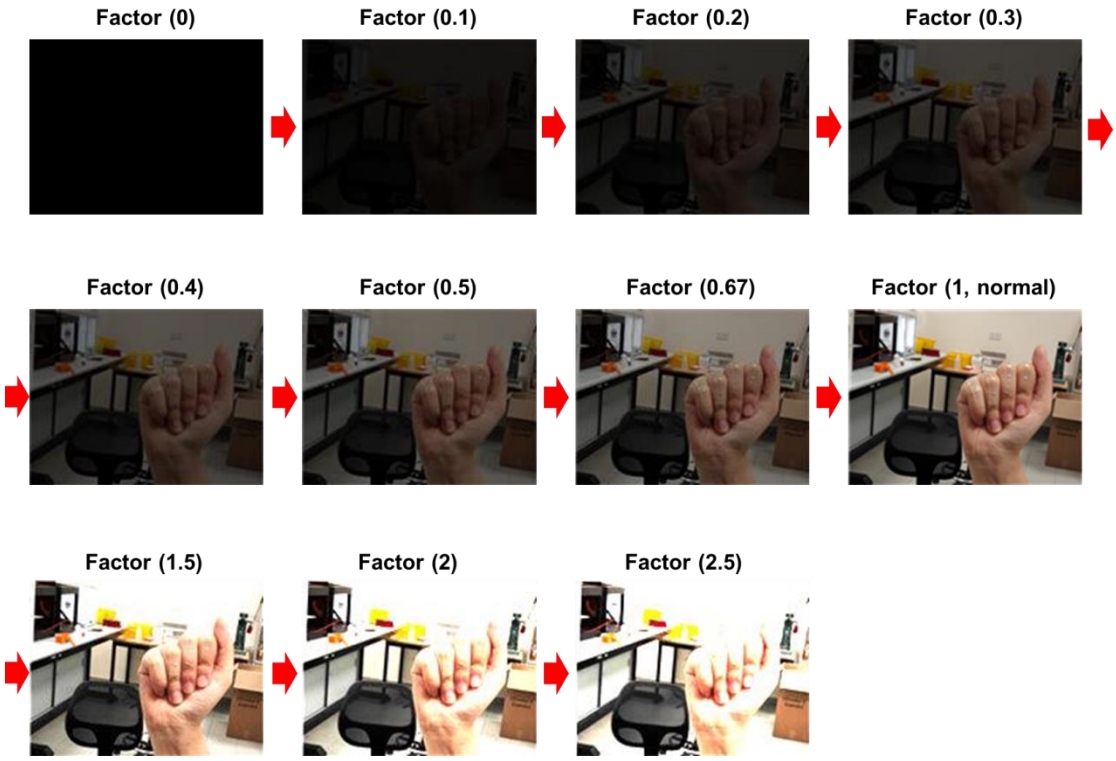
2: Camera——Visual information;

3: Computer ——BSV recognition system ;

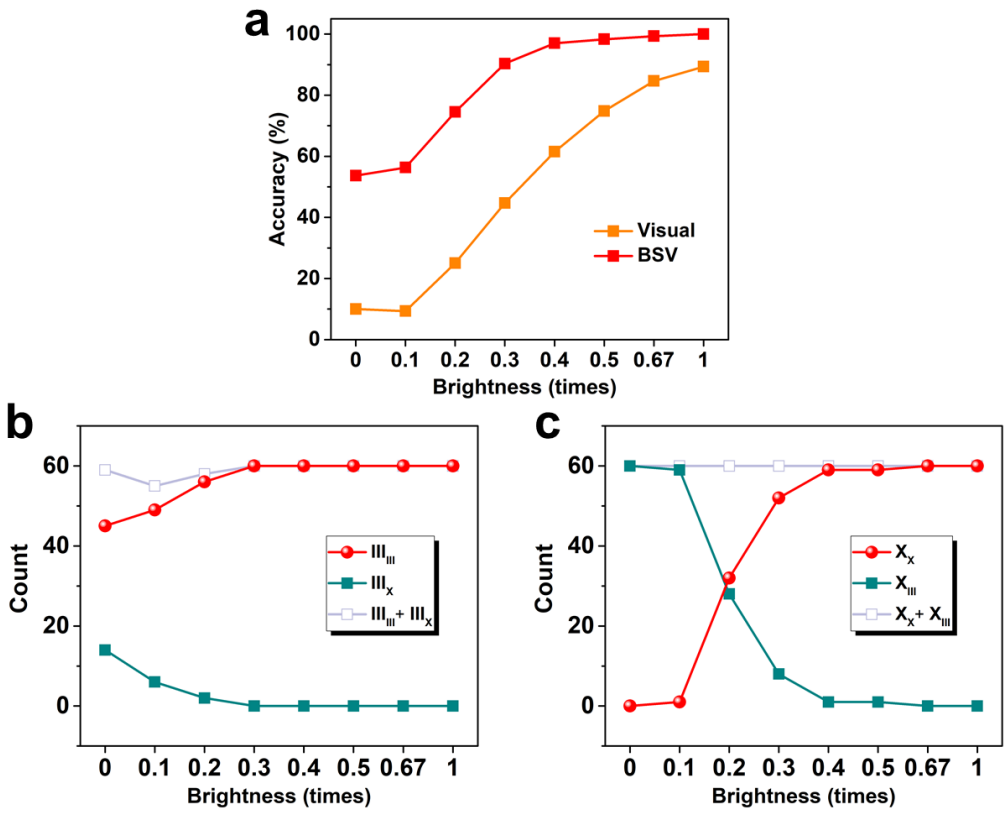
4: Wireless module —— data transmission ;

5: A quadruped robot.

Supplementary Figure 9 Photograph of auto-recognition and control system, consisting of a data acquisition unit (DAQ) for capturing the somatosensory information of a hand, a built-in camera of a computer for capturing the images of hand gestures, the computer for implementing the BSV associated learning, a wireless data transmission module, and a quadruped robot.



Supplementary Figure 10 Processing the normal testing dataset with the variation value of brightness. MATLAB function: hsv = rgb2hsv(testing_image); hsv(:, :, 3)*factor. The RGB values of testing images (visual information) were converted to the appropriate hue, saturation, and value (HSV) coordinates. Then, the value was multiplied by different factors (0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.67, 1.5, 2, and 2.5) to adjust the brightness of the image.



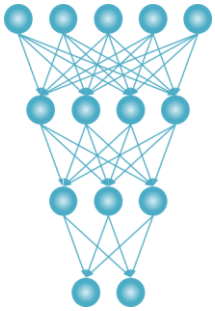
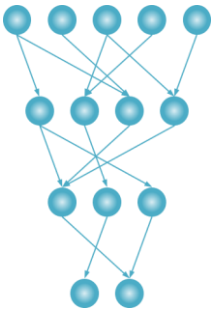
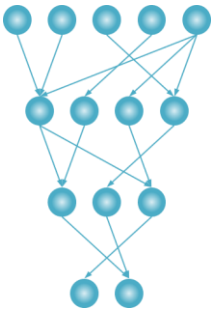
Supplementary Figure 11 (a) Overall recognition accuracy of the visual- and BSV-based recognition approaches under various dark environments. Individual recognition results of the specific hand gestures III (b) & X (c) under various dark environments. The symbol "A_B" represents the true hand gesture "A" is classified into the gesture "B" (predicted class). For example, the III_X means that a true hand gesture III is recognized into the category of X.

Note: From Fig. 3d and Supplementary Figure 7b, hand gestures III and X are easily misrecognized into each other using only the somatosensory data. However, under the harsh environments (brightness factors > 0.3, excluding the extreme cases (brightness factors < 0.3) in which the visual information does not work), the BSV associated learning still can correctly recognize almost all the hand gestures. In

other words, the BSV associated learning not only easily recognizes the hand gestures that can be well classified using only the somatosensory data, but also can well distinguish the hand gestures III and X. This result indicates that, even for the confused somatosensory information, it still can play a necessary role in the visual-somatosensory recognition including the harsh environments (defective visual information). It can be explained by the effective interaction and complementation of visual-somatosensory information in the BSV associated learning, resulting in the tolerance of dark environments.

III: Supplementary Tables

Supplementary Table 1 Comparison of the BSV associated learning with dense and sparse connectivity for fusion of visual and somatosensory features[1].

	Dense neural network	Sparse neural network	
		Pruning based on absolute value	Pruning based on Frobenius condition number
Architecture	BSV	BSV	BSV
Visual processing	CNN	CNN	CNN
Somatosensory processing	Normalization pre-processing	Normalization pre-processing	Normalization pre-processing
Fusion processing			
Pruning criterion	—	$\lvert w_{ij} \rvert < \varepsilon_1$	$\frac{\sigma_F(\tilde{W})}{\sigma_F(W)} \leq \varepsilon_2$
Network sparsity	—	0.436	0.436
Accuracy	**99.7%**	**97.8%**	**100%**

[1]Absolute value- and Frobenius condition number-pruning strategies are employed to achieve sparse neural networks, respectively. ε_1 and ε_2 are two user-defined thresholds for pruning criterion.

Supplementary Table 2 Comparison of the BSV associated learning with other methods, including multi-modal and multi-feature fusion[1].

Method	Modality	Accuracy (%)	Dark light tolerance (factor)	Noise tolerance (times)	Reference
BSV	Visual (camera) + Somatosensory (skin-like sensor)	**100**	**~0.29**	**~0.0158**	Our
SV-V	Visual (camera) + Somatosensory (skin-like sensor)	**93.7**	~0.57	~0.0028	
SV-T	Visual (camera) + Somatosensory (skin-like sensor)	**95.8**	~0.51	~0.0044	
SV-M	Visual (camera) + Somatosensory (skin-like sensor)	**97**	~0.46	~0.0042	
HMM fusion	Visual (Kinect sensor) + Wearable (inertial sensor)	**93**	NA	NA	S3
BayCoBoost	Visual (Kinect sensor) + Wearable (inertial sensor)	**97.6**	NA	NA	S4
Visual attention	Multiple visual features (shape + texture + color)	**94.36**	NA	NA	S5
	Multiple visual features (shape + texture)	**87.72**	NA	NA	
Saliency-guided method	Multiple visual features (image saliency + skin information)	**95.27**	NA	NA	S6
HOG-LBP fusion[2]	Multiple visual features (HOG + LBP features)	**97.8**	NA	NA	S7

[1]Here, the selected multi-feature fusion methods are based on a public hand gesture dataset - NUS Hand Posture dataset II (subset A)[S5], because the image data in this dataset are similar to those we used. The tolerance degree of dark light (noise) is defined as the brightness factor (Gaussian noise variance time) of visual data when the system recognition ability decreases 90% when using its normal visual information. For light condition, normal visual information means the brightness factor is 1; while for noise condition, normal visual information indicates the variance time is 0.

[2]Need to manually annotate hand regions.

Supplementary Table 3 Evaluation of the BSV associated learning on the public NUS Hand Posture dataset II (subset A)[1].

Method	Dataset		Accuracy (%)
	Visual	Somatosensory	
BSV	Our image dataset	Our strain sensor dataset	**100**
BSV	NUS-II dataset	Our strain sensor dataset	**99.3**
SV-V	NUS-II dataset	Our strain sensor dataset	**90.0**
SV-T	NUS-II dataset	Our strain sensor dataset	**96.8**
SV-M	NUS-II dataset	Our strain sensor dataset	**93**

[1]In the multimodal somatosensory-visual (SV) dataset, visual data are the images from the NUS Hand Posture dataset II (subset A) while somatosensory data are provided by our original strain sensor dataset. To prepare the SV dataset, we cut our strain sensor data list (3000 points) into a new strain sensor data list (2000 points) as the corresponding somatosensory information of the NUS-II subset A. In detail, for each category of hand gestures (from I to X), the last 100 points was directly removed. The remaining strain sensor data of 10 categories (200 points for each category) are combined into a new strain sensor data list. Without any amendment of strain sensor data, a new multimodal SV dataset (regarding as an amended public dataset) can be constructed based on the public NUS-II subset A.

IV: Supplementary References

S1. Han, S., Mao, H. & Dally, W.J. in *International Conference on Learning Representations* (2016).

S2. Le, X. & Wang, J. Robust pole assignment for synthesizing feedback control systems using recurrent neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **25**, 383-393 (2013).

S3. Liu, K., Chen, C., Jafari, R. & Kehtarnavaz, N. Fusion of inertial and depth sensor data for robust hand gesture recognition. *IEEE Sens. J.* **14**, 1898-1903 (2014).

S4. Wu, J. & Cheng, J. Bayesian co-boosting for multi-modal gesture recognition. *J. Mach. Learn. Res.* **15**, 3013-3036 (2014).

S5. Pisharady, P.K., Vadakkepat, P. & Loh, A.P. Attention based detection and recognition of hand postures against complex backgrounds. *Int. J. Comput. Vis.* **101**, 403-419 (2013).

S6. Chuang, Y., Chen, L. & Chen, G. Saliency-guided improvement for hand posture detection and recognition. *Neurocomputing* **133**, 404-415 (2014).

S7. Zhang, F., Liu, Y., Zou, C. & Wang, Y. in 2018 *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)* 1-6 (2018).