Robotica: (2021) volume 39, pp. 378–390. © The Author(s), 2020. Published by Cambridge University Press.
doi:10.1017/S0263574720000429

Soft Robotic Finger Embedded with Visual Sensor for Bending Perception

Shixin Zhang†, Jianhua Shan†, Bin Fang‡* and Fuchun Sun‡*

†*College of Mechanical Engineering, Anhui University of Technology, Maanshan 243002, China*
E-mails: zsx1723190077@163.com, 379751793@qq.com
‡*Department of Computer Science and Technology, Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China*

(Accepted May 3, 2020. First published online: June 17, 2020)

SUMMARY

The various vision-based tactile sensors have been developed for robotic perception in recent years. In this paper, the novel soft robotic finger embedded with the visual sensor is proposed for perception. It consists of a colored soft inner chamber, an outer structure, and an endoscope camera. The bending perception algorithm based on image preprocessing and deep learning is proposed. The boundary of color regions and the position of marker dots are extracted from the inner chamber image and label image, respectively. Then the convolutional neural network with multi-task learning is trained to obtain bending states of the finger. Finally, the experiments are implemented to verify the effectiveness of the proposed method.

KEYWORDS: Soft robotic finger; Visual sensor; Multi-task learning; Convolutional neural network.

1. Introduction

Nowadays, the robotic hands are gradually required to be highly flexible and absolutely security for the human–robot interaction. The traditional rigid robotic hand[1,2] hardly meets the requirements. However, the soft robotic hand has remarkable advantages in the operation of fragile objects and in collaboration with human beings. Additionally, they have several advantages, such as lightweight, inexpensive, easily fabricated, and simply controlling.

In order to determine real-time states of the soft hand, the sensors are essential. However, due to the large deformation of the soft hand, the traditional sensing devices are easily damaged or difficult to fit closely with the finger. Therefore, the flexible sensors, which can adapt to the bending and stretching characteristics of the soft hand, are gradually developed. The most commonly used flexible sensors are resistive sensors,[3,4] whose resistance of specific material changes linearly as bending. Besides, optical fiber, magnet, and hall sensors are also used to measure the bending degree of soft finger.[5,6] However, the above sensors have the same limitation that they only measure bending curvature in single bending direction.

The visual sensor has wide vision and high resolution. Hence, it would get more information such as extrusion deformation and bending state of the soft finger. There have been many studies on the acquisition of contact force, texture, and other information based on visual sensor.[7–9] The sensors consist of a CCD camera, acrylic plate, and elastic body. On the surface of elastomers, some markers are added at random distribution or uniform distribution. Based on the displacement of

Shixin Zhang and Jianhua Shan contributed equally to this work.

* Corresponding author. E-mails: fangbin@mail.tsinghua.edu.cn; fcsun@tsinghua.edu.cn

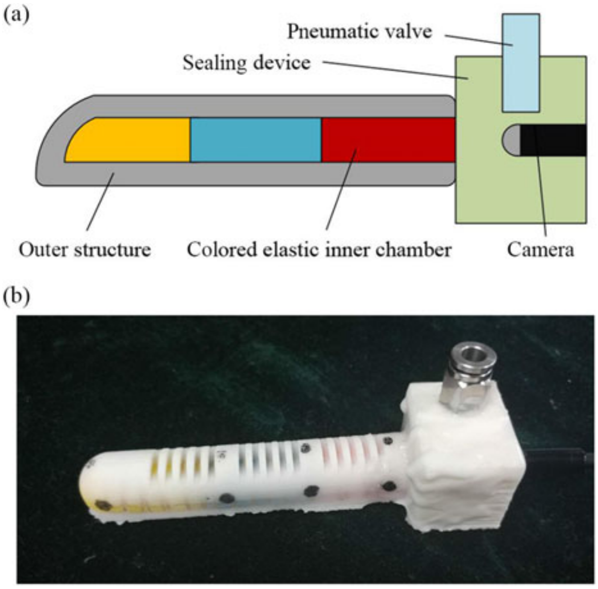

Fig. 1. The proposed soft finger with visual sensor. (a) Structure of the proposed soft finger. (b) Prototype.

the markers, the detection of tactile and the distribution of force can be obtained.[10,11] While some researchers focused on the detection of slip, shear force,[12,13] and texture recognition,[14,15] besides the other researchers studied on perceiving edge of object,[16–18] which used pins to replace markers in sensors. However, the visual sensors mentioned above are mostly fixed on the fingertip of the rigid hand, which means the perception area is limited to the fingertips. Therefore, in order to apply visual sensors in the soft hand, it is necessary to change the conventional sensing mode of visual sensors. In this paper, a novel visual sensing mode is proposed. The elastomers are made as the inner chamber of soft finger with three colors. The bending states of the soft finger are obtained by recognizing the changing of the color regions.

The rest of this paper is organized as follows. Design of soft robotic finger is described in Section 2, while Section 3 introduces our proposed perception algorithm. Section 4 explains experiment scheme and results. Finally, future work and conclusions are described in Section 5.

2. Soft Robotic Finger Design

The proposed soft robotic finger consists of an outer structure, a colored inner chamber, a sealing device, an endoscope camera, and a pneumatic valve. The outer chamber has a certain hardness to support the soft finger. The inner chamber is relatively soft to realize the pneumatic bending function of the finger. Meanwhile, the inner chamber is used for sensing by identifying different color regions. Based on the inspiration of human fingers, the soft finger is divided into three sections. In order to distinguish each section and facilitate image segmentation, different colors are adopted. Three colors (red, blue, and yellow) are used to present different sections. Referring to the color selection, the RGB values of color regions require to be distinctly different for identification. For example, when red channel value is highest, color is absolutely recognized as red. Besides, the inner chamber is in close contact with the outer chamber to prevent expansion cracking. When the soft finger is bending, the camera embedded into the sealing device is used to capture the images of the deformation of the inner chamber. The camera's resolution is 640 × 480 and the frame rate is as high as 30 fps. Referring to human's finger, the length of inner chamber is selected as 10 cm. And the length of each color region is divided as follows: 4 cm for red region, 3 cm for blue region, and 3 cm for yellow region. The proposed soft finger is shown in Fig. 1.

3. Algorithm

In this section, the perception algorithm based on image preprocessing and deep learning is proposed. We extract boundary of color regions from inner chamber image and get position of marker dots from label image by image preprocessing. The bending state of finger is predicted from boundary by convolutional neural network (CNN). The proposal of the algorithm is shown in Fig. 2.

https://doi.org/10.1017/S0263574720000429 Published online by Cambridge University Press

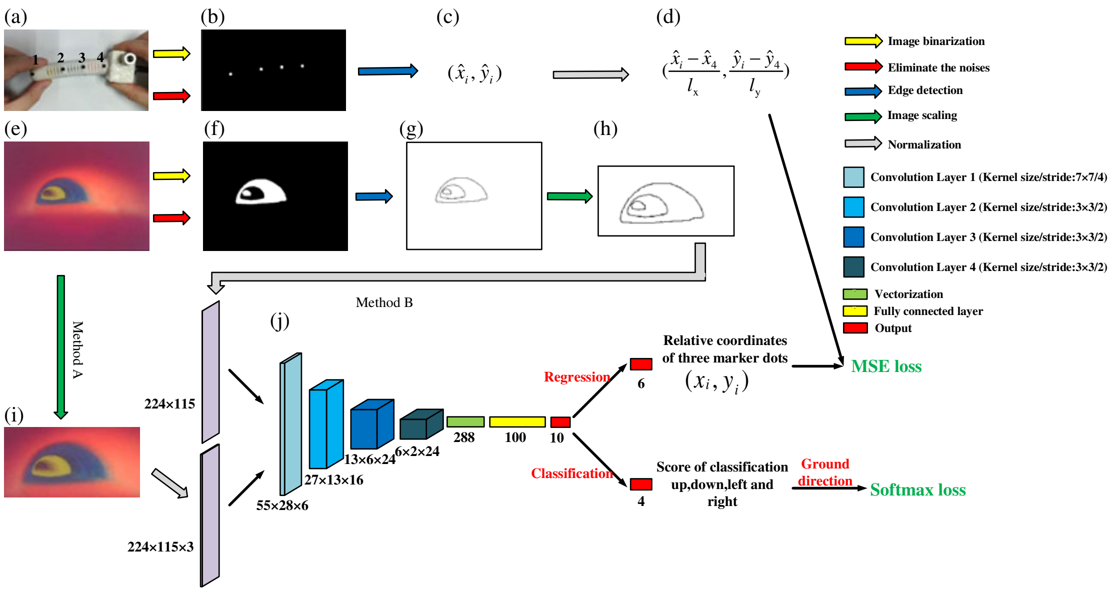
(a)
1 2 3 4
(b)
(c)
$(\hat{x}_i, \hat{y}_i)$
(d)
$(\frac{\hat{x}_i - \hat{x}_4}{l_x}, \frac{\hat{y}_i - \hat{y}_4}{l_y})$
Image binarization
Eliminate the noises
Edge detection
Image scaling
Normalization
Convolution Layer 1 (Kernel size/stride:7×7/4)
Convolution Layer 2 (Kernel size/stride:3×3/2)
Convolution Layer 3 (Kernel size/stride:3×3/2)
Convolution Layer 4 (Kernel size/stride:3×3/2)
Vectorization
Fully connected layer
Output
(e)
(f)
(g)
(h)
Method A
Method B
(i)
(j)
224×115
224×115×3
55×28×6
27×13×16
13×6×24
6×2×24
288
100
10
Regression
6
Relative coordinates of three marker dots
(x_i, y_i)
MSE loss
Classification
4
Score of classification up,down,left and right
Ground direction
Softmax loss

Fig. 2. The overview of algorithm flow. (a) Label image is captured by outer camera. Four dots are marked on the surface of finger. (b) Marker dots are extracted after image binarization. (c) Coordinates of marker dots are obtained from (b). (d) The fourth marker dot is unchanged. We select it as reference marker dot and realize coordinates normalization. (e) For original inner chamber image that is captured by embedded camera, we present two methods to process. Method B is introduced first: (f) Color regions are obtained via binarization. (g) Boundaries of color regions are extracted from (f). (h) Process boundary image with segmentation and scaling. The size is reduced from 640 × 480 to 224 × 115. In this step, method B is finished. (i) Method A is that directly process original inner chamber image referring to step (h). Different from method B, method A get boundary via CNN, which is possible to be affected by difference of pixel values. (j) Two methods share the structure of CNN. We adopt multi-task learning to realize regression task and classification task at same time. Loss weight is appended to adjust training difficulties of tasks.

3.1. Image preprocessing

The inner chamber image is captured by embedded camera to infer bending state of finger. The pixel values of color regions cannot represent the bending state of finger effectively. However, the boundary of color regions can represent the bending state. The size of boundary reflects the bending degree of finger, and the relative position of boundary reflects bending direction. In order to get boundary, we cut out and scale the inner chamber image. Then, CNN is used to get boundary information (method A). However, due to the difference of pixel values of same color region, they are possible to mistake as boundary. Therefore, we present another method that boundary is extracted manually by image preprocessing before using CNN (method B). The image preprocessing is as follows:

(1) *Image binarization*. Due to obvious difference between RGB values, the image can be divided into different regions by setting threshold value. Thus, the original image is turned into binary image completely.
(2) *Denoising*. There are still noises in the image after the step 1. In order to get rid of the noises, dilatation and corrosion transformation is performed on the binary image with 3 × 3 core twice.
(3) *Edge detection*. Using the edge detection function of OpenCV, the boundary of each color region is obtained.
(4) *Image segmentation and scaling*. The large image would increase the number of network parameters. Therefore, the boundary image requires segmentation and scaling from 640 × 480 to 224 × 115.

Meanwhile, we use positions of joints and fingertip, referring to human finger (see Fig. 3(a)). Four marker dots are used to represent joints and fingertip (see Fig. 3(b)), which can fit bending curve of finger. The positions of marker dots are captured by outer camera as label image, which is processed in above steps as well.

https://doi.org/10.1017/S0263574720000429 Published online by Cambridge University Press

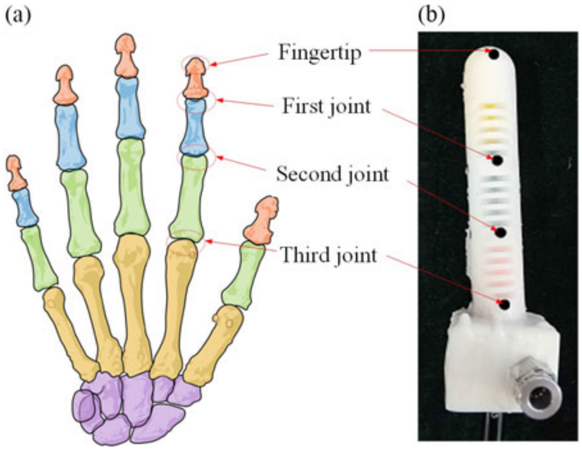

Fig. 3. The soft robotic finger referring to human finger. (a) Prototype of the human finger. (b) Prototype of the proposed soft robotic finger. Referring to human finger, marker dots represent the joints and fingertip. Via marker dots, we can fit bending curve to represent finger state.

3.2. Algorithm of CNN

After image processing, the processed images require to be normalized. For method A, the pixel values of scaled inner chamber image divide by 255. For method B, the pixel values of boundary (see Fig. 2(h)) are set as 1 and the others are set as 0, which is as normalization equivalently. Besides, the form of normalization of marker dots coordinates is as follows:

$$(x, y) = \left(\frac{x - x_{re}}{l_x}, \frac{y - y_{re}}{l_y} \right) \tag{1}$$

where (x_{re}, y_{re}) is coordinate of reference marker dot. l_x and L_y are scale coefficient. Via normalization, convergence speed is faster and errors are decreased effectively.

Then CNN[19,20] is applied for recognizing the inner chamber image. Because the features of the boundary information are obvious, so the simple CNN can easily recognize them. The structure of CNN is shown in Fig. 3(j). Each convolutional layer is followed by an activation layer, where rectified linear unit (ReLU)[21,22] is selected as activation function instead of Sigmoid[22] and Tanh[23] function. Since their gradient in the saturated region is close to 0, which is easy to cause the problem of vanishing gradient and reduce the convergence speed. On the contrary, ReLU's gradient is equal to 1 when value is more than 0, which is helpful to solve the convergence problem of network.

In order to realize classification and regression, different classifiers and loss functions are used. Softmax[24] is selected as classifier to recognize directions (up, down, left, and right). The classification output is more intuitive with the form of the normalized probability. Before softmax, each direction score has been computed firstly. Hence assume score vector is $s = (s_1, s_2, s_3, s_4)$. While some scores are possibly negative so that they cannot reflect relative size. It is precisly because of characteristic of exponential function, whose value is invariably positive and monotonically increasing. We make score vector map to it ($e_s = (e^{s_1}, e^{s_2}, e^{s_3}, e^{s_4})$). The definition of softmax is as follows:

$$f_i(s) = \frac{e^{s_{yi}}}{\sum_{j=1}^{4} e^{s_j}} \tag{2}$$

where s_{yi} is score of ground category of ith sample, while s_j represents score of all categories of ith sample.

For the ground category, its score is supposed to be highest. When softmax output is less than 1, it requires to be punished. Furthermore, for smaller output, punishment requires to be increased largely, which means loss is boosted largely as well. Therefore, the negative logarithm is used to meet these requirements. The softmax loss function[24] is defined as:

https://doi.org/10.1017/S0263574720000429 Published online by Cambridge University Press

$$L_c = -\log \frac{e^{s_{yi}}}{\sum_{j=1}^{4} e^{s_j}} \tag{3}$$

Via regression task, the predicted coordinates of marker dots can be estimated. Because the fourth marker dot is fixed, it can be set as reference dot. Hence, we only require relative coordinates of the other three marker dots. Different with softmax loss, regression loss consists of the difference between the ground and predicted values. The specific form is as follows:

$$L_r = \sum_{i=1}^{3} L_{ri} = \sum_{i=1}^{3} \{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\} \tag{4}$$

where L_r and L_{ri} are called mean square error[25] loss function. L_{ri} represents regression loss of each marker dot. (x_i, y_i) is predicted relative coordinates, and $(\hat{x}_i, \hat{y}_i)$ is ground relative coordinates.

If different CNN is used to respectively realize classification and regression task, it would waste the computing resources and increase run time. Thus, we adopt multi-task learning to simultaneously realize both tasks, which share one network. The total loss is the weight sum of the classification and regression loss.

$$L_t = \beta_i \sum_{i=1}^{3} L_{ri} + \lambda L_c \tag{5}$$

where β_i and λ are loss weights.

The loss weights are determined by difficulty degree of tasks. Compared with regression task, classification task is easier. So λ is less than β_i. On the regression task, marker dots locate different distance from reference marker dot. So the distance is further, the changing range is wider, and the learning difficulty is higher. Therefore, the relation of loss weight is $\beta_1 > \beta_2 > \beta_3 >> \lambda$.

For optimizer selection, Batch Gradient Descent (BGD),[26] Stochastic Gradient Descent (SGD),[27] Momentum,[28] Nesterov Accelerated Gradient (NAG),[29] Adaptive Gradient Algorithm (Adagrad),[30] An Adaptive Learning Rate Method (AdaDelta),[31] RMSprop,[32] and Adaptive Moment Estimation (Adam)[33] are common optimizer. Compared with BGD, SGD, Momentum, and NAG whose learning rate is set manually, the others adopt adaptive learning to promote network convergence faster. In addition, Adam combines the advantages of RMSprop, Momentum, and Adagrad to achieve better adaptive learning. To sum up, we prefer to select Adam.

3.3. Bending state presentation

Via regression task, the positions of three marker dots are determined. Then four dots including fixed reference marker dot represent finger state with Bẽzier curve.[34] Bẽzier curve is a parametric curve frequently used in computer graphics and related fields, which can be fit by four points that are the starting point p_0, the ending point p_1, and two separate intermediate points p_2 and p_3. The parametric form of the curve is as follows:

$$B(t) = (1-t)^3 p_0 + 3t(1-t)^2 p_1 + 3t^2(1-t) p_2 + t^3 p_3, \; t \in [0, 1] \tag{6}$$

4. Experiments

In order to verify the effectiveness of the proposed algorithm, the data sets are built and the experiments are designed.

4.1. Data sets

We collect data sets in four bending directions. The inner chamber images in down bending direction are taken as example. Figure 4(a) shows original inner chamber images in different bending states. The data sets consist of scaled inner chamber images (see Fig. 4(b)) via method A or boundary images (see Fig. 4(c)) via method B, positions of marker dots (see Fig. 4(d)), and four bending directions (up, down, left, and right). In order to collect enough data at the same time, we record the bending process forming as videos via embedded and outer cameras, which are later cut into images frame-by-frame. We use 2200 samples, 10% of them are used as test samples.

https://doi.org/10.1017/S0263574720000429 Published online by Cambridge University Press

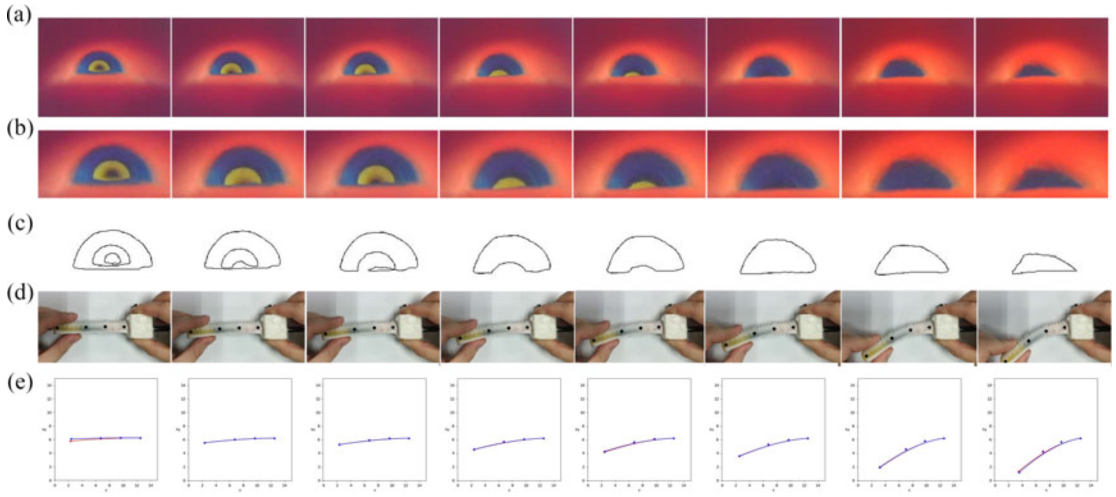

Fig. 4. Example of data sets of two methods in down bending direction. (a) The original inner chamber images. (b) The scaled inner chamber images are obtained via method A. (c) The boundary images are obtained via method B. We enlarge (b) and (c) images to show clearly. Their actual size is 224 × 115. (d) The label images are captured during down bending, which includes positions of marker dots. (e) The predicted marker dots are fitted with Bĕzier curve to represent bending curve. Blue and red curves represent the ground and predicted bending curves, respectively. In the first figure of (e), there are some errors in the first marker dot. In the other figures, the predicted curves are so close to ground curves.

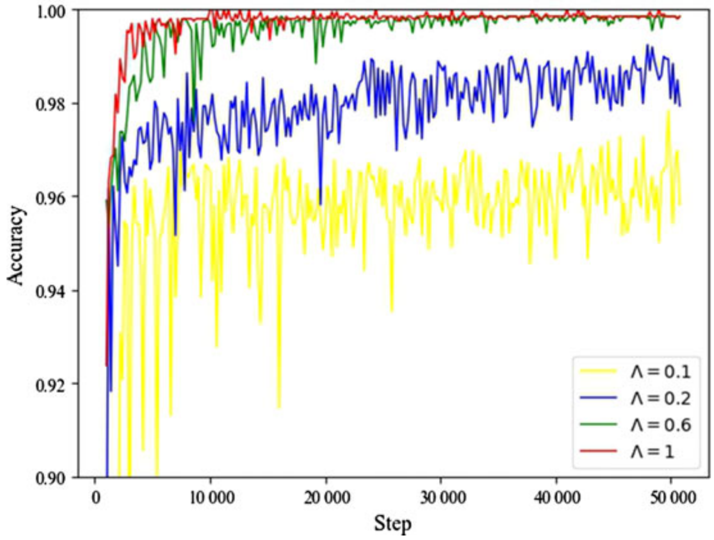

Fig. 5. Relationship between accuracy and loss weight λ. When the loss weight β_i are fixed, as increasing λ, the accuracy is boosted. However, when λ is up to a certain value, effect tends to be saturated. Additionally, the larger λ is possible to affect regression task.

4.2. *Classification and regression*

The scaled inner chamber images are selected firstly. The CNN is built with Pytorch 1.0 (Facebook AI Research, Menlo Park, California, USA) and Python 3.6 (Python Software Foundation, Beaverton, Oregon, USA). In Pytorch, weight initialization is realized automatically. Due to select Adam optimizer, learning rate is recommended as 0.003, which has verified that the effect of this learning rate for Adam is best in numerous test experiments. The initial regularization weight is set as 0.01, we can adjust values based on training situation. For loss weight λ, we select different values to test effect (see Fig. 5). As increasing λ, the maximum accuracy is increased and tends to stabilize. When $\lambda = 1$, the maximum accuracy is steadily close to 100% and accuracy curve is smooth. The accuracy of each direction is shown in Fig. 6.

https://doi.org/10.1017/S0263574720000429 Published online by Cambridge University Press

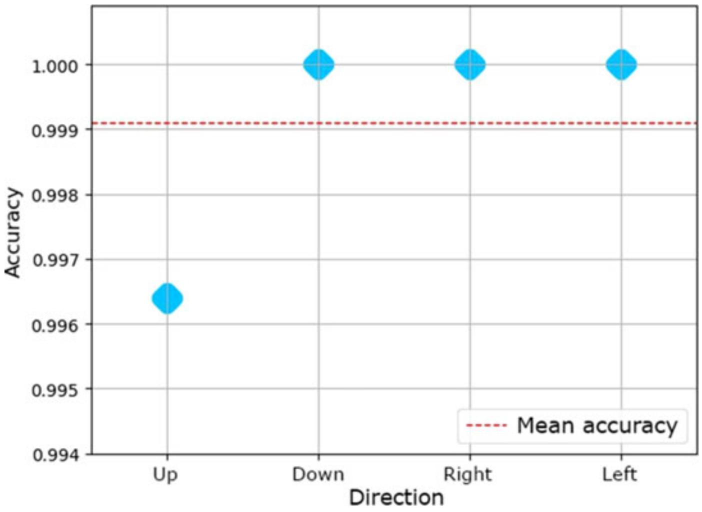

Fig. 6. Accuracy distribution of each bending direction.

For regression task, scheme A ($\beta_i = 1, 1, 1; \lambda = 1$) is used to experiment firstly. Via training, Fig. 7 reflects distribution of marker dots in different directions. When bending up and down, bending does not happen in x-axis direction. In order to visualization, we transform the distribution of marker dots from three-dimensional coordinate to y–z plane. Similarly, in the left and right directions, the distribution is shown in x–y plane. Figure 7(a) and (b) shows that the degree of dispersion of predicted dots is larger, especially the first marker dots are distinct than the others. In Fig. 8, the bar chart and error bars reflect the mean and distribution of the errors of the marker dots. In order to decrease the errors of marker dots, loss weight is adjusted as 10, 5, 3, and 1 (scheme B).

As shown in Fig. 7(c) and (d), the degree of dispersion of the first predicted dots is significantly decreased, which are closer to the ground dots. Besides, the mean errors of the first marker dot are decreased about 50% (see Fig. 8), while the errors of the other marker dots are decreased in different degrees.

4.3. Experimental results

We get acceptable results by the method A. However, the errors of marker dots are expected to decrease further. Because manually extracting boundary in method B can improve boundary precision to decrease recognition errors. In order to test method B, similarly, we select 2200 boundary images, where 10% images are as test sets. The structure of CNN and the most parameter settings are unchanged, except for channel parameter because the channel of boundary image is one rather than three. In Fig. 9, obviously, the errors of marker dots are decreased further. Besides, the ranges of errors are reduced as shown in error bar. This suggests that the precise boundary information improves effect of recognition. Thus, compared with method A, method B is more effective.

In order to test robustness of two methods, we collect samples in different light environments. As shown in Fig. 10(a), the inner chamber images are captured in the initial states of finger, which are in the dark, a bit dark, common, and bright environment, respectively. The red and blue represent predicted and ground values of dots, respectively, as shown in Fig. 10(b) and (c).

The results show that the predicted marker dots with method B are closer to ground marker dots than method A in the conditions of bright, common, and a bit dark. In the bright and a bit dark environment, pixel values of same color region have huge difference, which results in mistaking to recognize as boundary with method A. On the contrast, method B avoids these mistakes via extracting boundary manually because method B does not care about the size of RGB values but concerns with the size relationship between each channel. For example, red only satisfies that the values of red channel are larger than the others, regardless of values size. Because boundary is extracted with some deviations in different environments, it is reasonable that there are a few errors in method B. However, in the dark environment, low brightness has serious influence on extracting boundary. In Fig. 11(d), the boundary of yellow region is divided into several parts, and boundary lines of each

https://doi.org/10.1017/S0263574720000429 Published online by Cambridge University Press

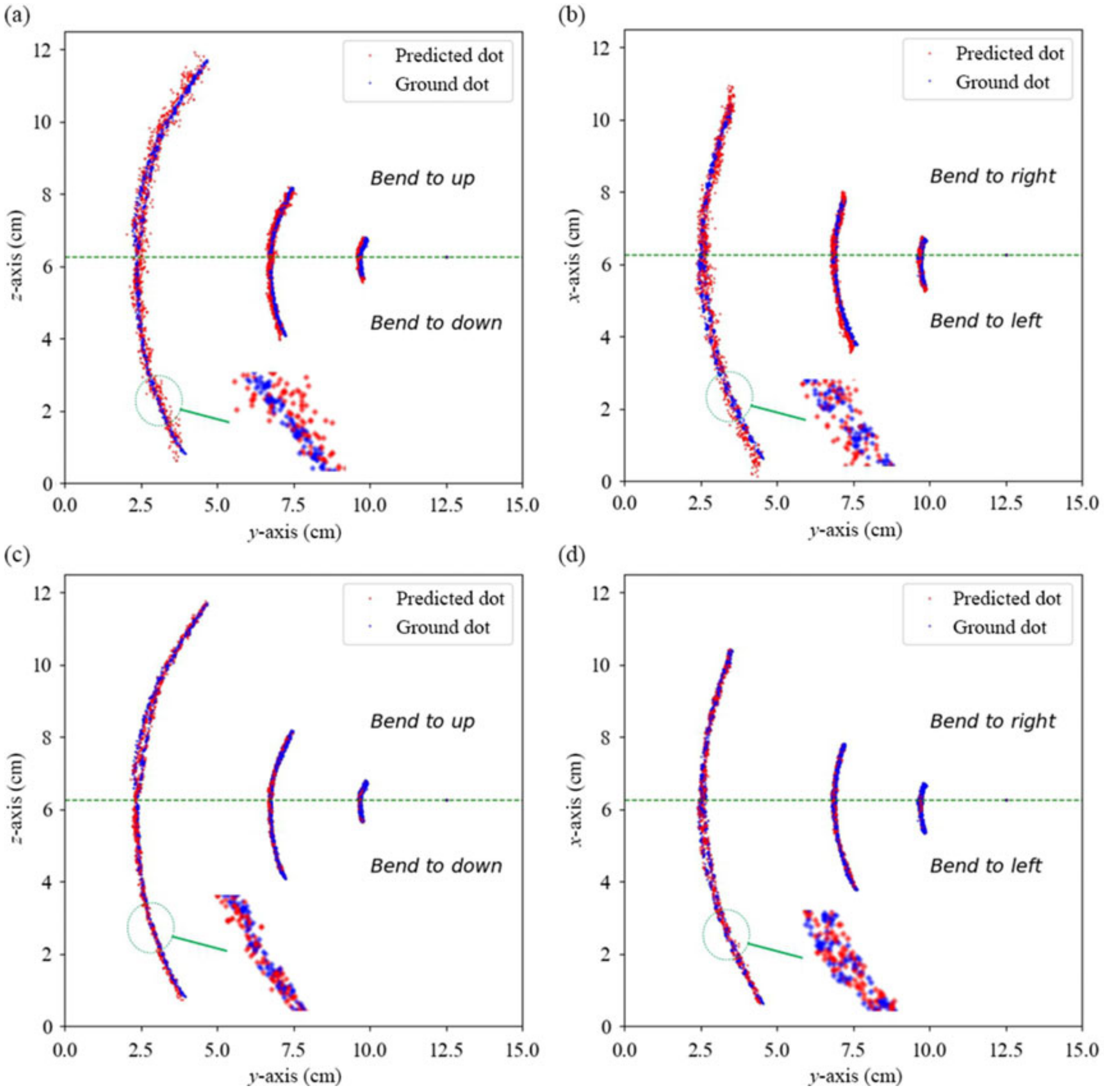

Fig. 7. Distribution of marker dots in different bending directions. We present distribution in a two-dimensional plane. The coordinate system represents relative location of marker dots. (a) and (b) The result of Scheme A, whose loss weights are all set as 1, shows large dispersion degree of predicted dots. (c) and (d) We present scheme B that adjusts loss weight as 10, 5, 3, and 1. It is found that the dispersion degree is decreased significantly.

region are rough. Because pixel values in the dark are small and in proximity to each other, resulting in that the relationship of each channel does not match the ground color. As shown in part of green circle in Fig. 11(d), some regions are shown as yellow, but the values of blue channel are the largest. Thus, some parts of yellow regions are recognized as blue. Besides, there are many other similar reasons resulting in errors, such as unsmooth boundary lines. While the effect of method A is better than method B in the dark environment, the difference of the small pixel values is tiny. In Fig. 11, it is found that the brightness is more adequate, boundary is clearer to extract. In order to reduce brightness interference, outer light can be appended to adjust environment brightness. Via above experiments, errors of marker dots are decreased effectively. With Bẽzier curve, we can get bending curve accurately. As shown in Fig. 4(e), the predicted curve and the ground curve are nearly the same. Due to errors of some first marker dots, the predicted bending curves have a few difference from ground bending curves, as shown in first figure of Fig. 4(e). Mostly, bending curves are predicted accurately. We adopt a better CPU (Intel core i7-8700) to test efficiency. According to 1000 tests, the result shows run time of CNN focus on 0–15 ms. The run time of image processing is about 10 ms. The sum of them ranges between 0 and 25 ms. For the camera (due to increase in imaging quality,

https://doi.org/10.1017/S0263574720000429 Published online by Cambridge University Press

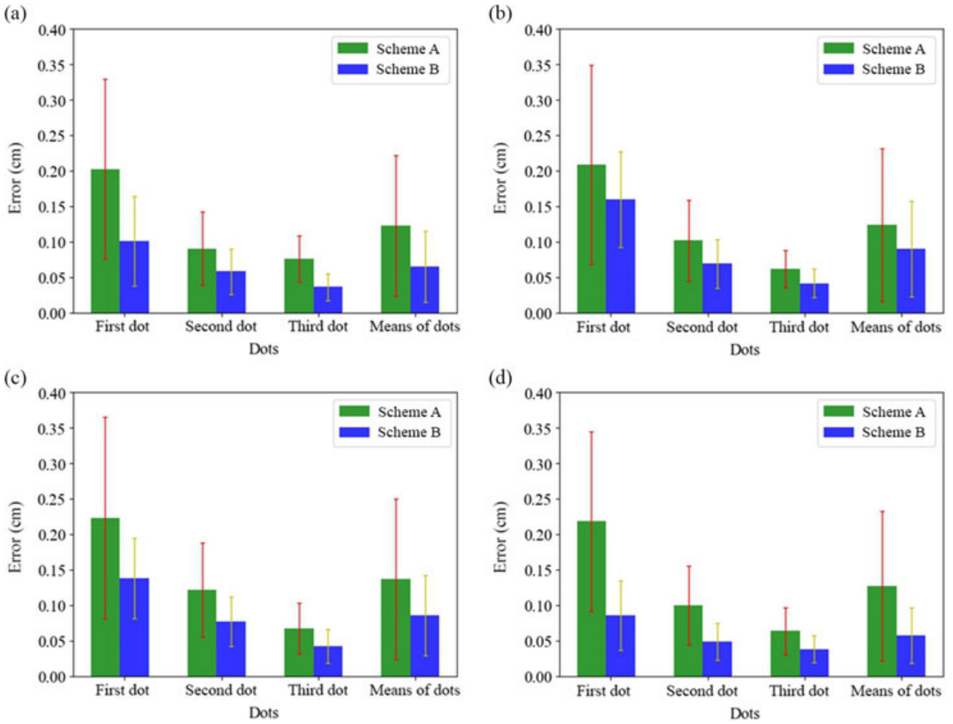

Fig. 8. Contrast of scheme A and B about distribution of mean errors of marker dots in different bending directions. (a)–(d) respectively show errors distribution in bending direction of up, down, left, and right. The blue bars show mean errors with loss weight $\beta_i = 10, 5, 3$ and $\lambda = 1$, and green bars show mean errors with loss weight $\beta_i = 1, 1, 1$ and $\lambda = 1$. The error bars are calculated with standard deviation. As loss weight increased, the errors decreased.

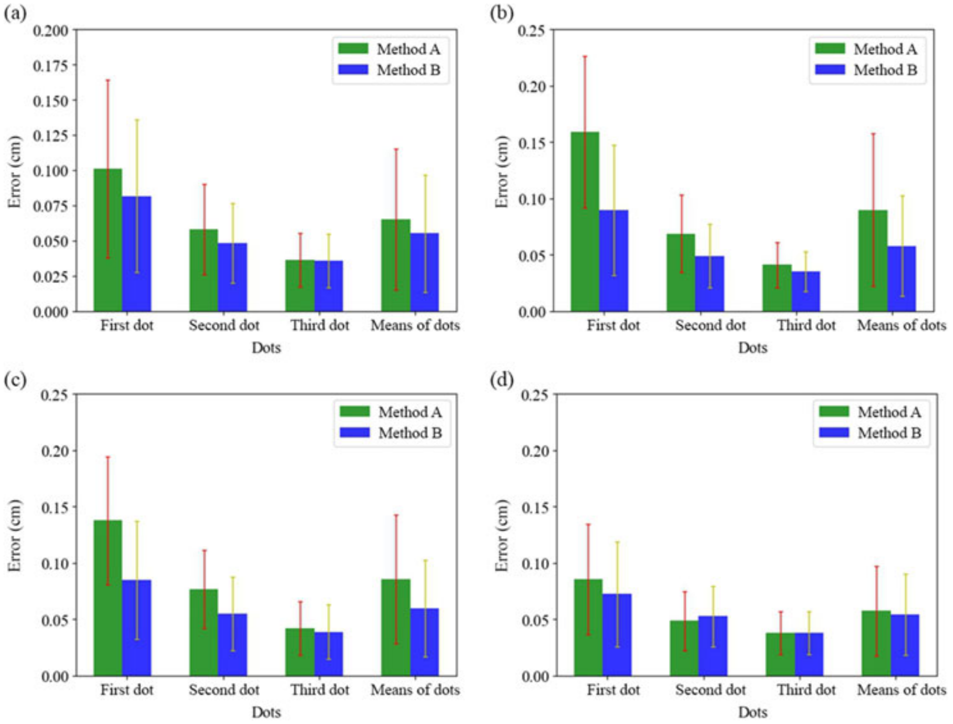

Fig. 9. Contrast of method A and B about distribution of mean errors of marker dots in different bending directions. (a)–(d) respectively show errors distribution in bending direction of up, down, left, and right. The green bars represent the mean errors with method A and blue bars represent mean errors with method B. The result shows that the difference of RGB values affects recognition precision of method A.

https://doi.org/10.1017/S0263574720000429 Published online by Cambridge University Press

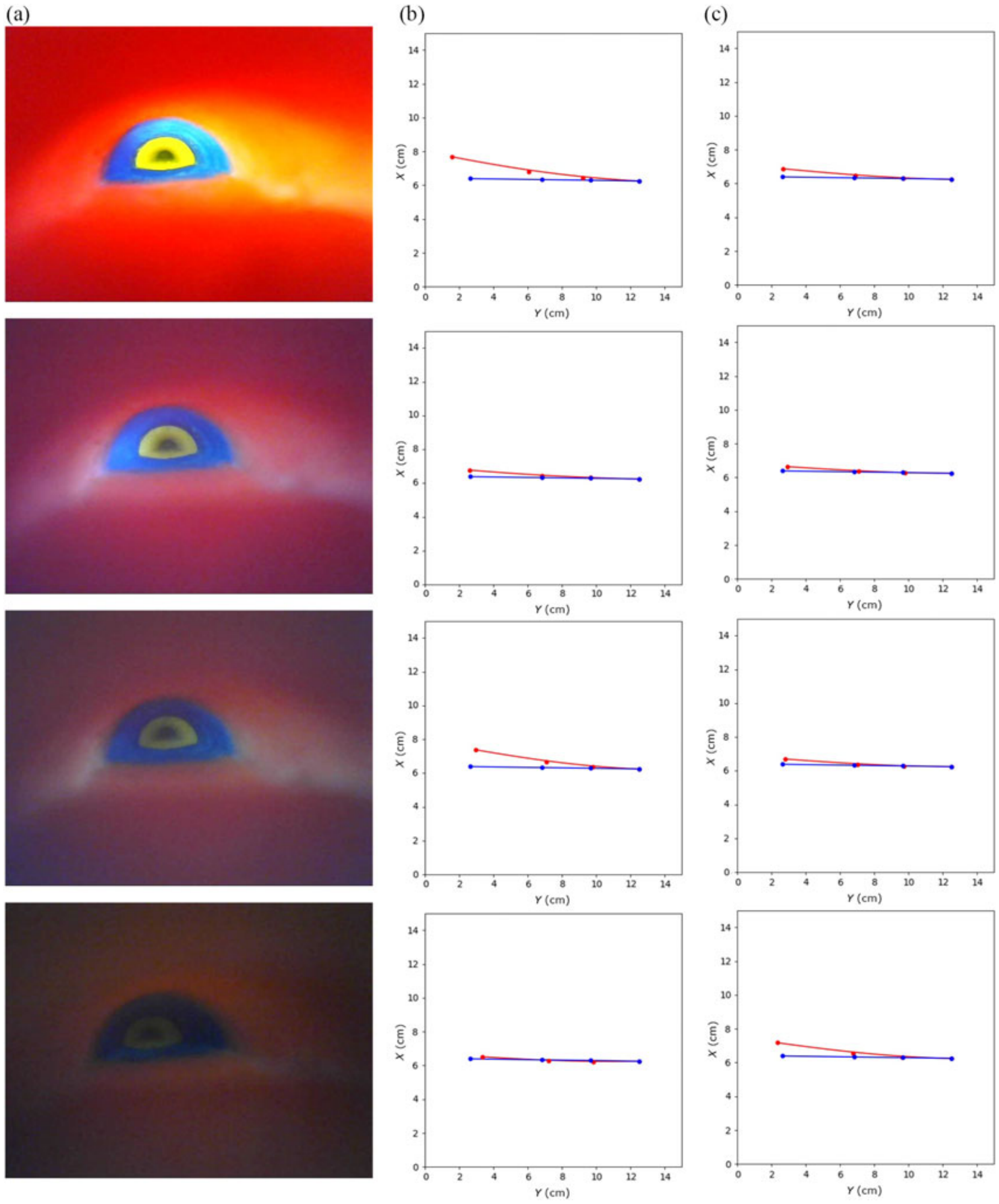

Fig. 10. The results of fitting bending curves with method A and B in different environments. (a) We select the conditions of bright, common, a bit dark, and dark to test robustness of two methods. The inner chamber images of initial state are captured as test data. (b) Represent the results of fitting bending curves with method A. There are larger errors in bright and a bit dark environment. (c) Represent the results of method B to fit bending curves. There are more errors appearing in the dark.

fps is 16), the interval between the previous frame and the next frame is about 50–65 ms, which is larger than the sum of run time of CNN and image processing. It means that the algorithm can realize real-time recognition.

4.4. Discussion

The classification accuracy can be close to 100%. With loss weight, the errors and dispersion degree are decreased effectively. Due to that the length of the soft finger is 10 cm and the mean errors of

https://doi.org/10.1017/S0263574720000429 Published online by Cambridge University Press

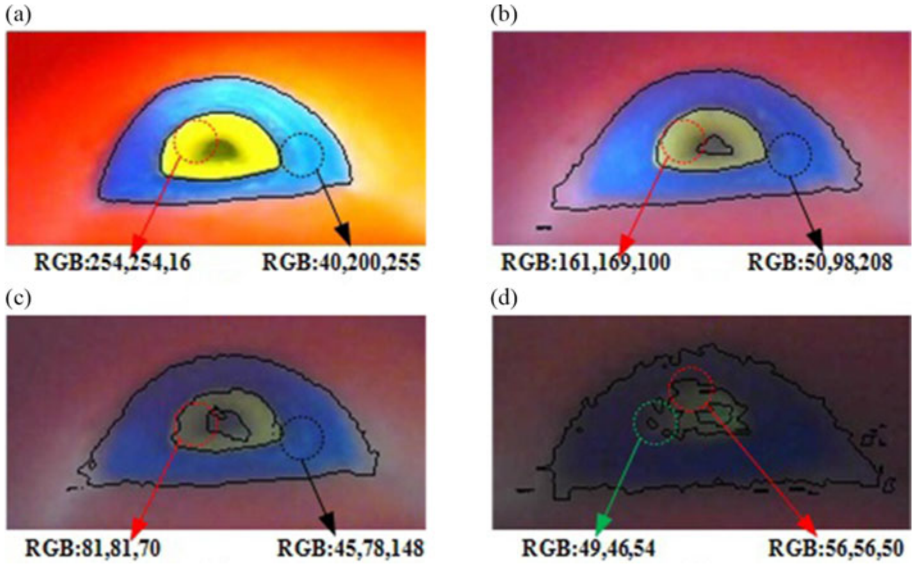

Fig. 11. Extracted boundary in different brightness environments. (a) Boundary is accurately extracted in bright environment. Because the RGB values are obviously different between each RGB channel. (b) In common environment, the RGB values are decreased resulting in that the difference is diminished. (c) In a bit dark environment, some channel values start to approach. The boundary lines are gradually unsmooth and errors of extracted boundary are increased. (d) Some regions are mistaken to recognize in the dark environment. As shown in part of green circle, the value of blue channel is the largest so that yellow is recognized as blue incorrectly. And the boundary is terribly rough.

marker dots are about 0.05 cm, which is reasonable. The results show that errors are mainly contributed by the first marker dots. From the distribution range of marker dots, the first marker dots have the maximum bending range, whose probability of error is naturally higher than other dots. According to the contrast experiment, method B can overcome different environment brightness via appending outer light. Besides, we can append noise in the captured inner chamber images randomly, which is helpful to increase recognition robustness.

5. Conclusions

In this paper, we design the perception algorithm for soft finger that is integrated with visual sensor. The inner chamber image has abundant bending character information, which can be used without complicated processing. Based on multi-task learning, the bending state is acquired accurately. In order to improve recognition effect of CNN, loss weights are appended. Additionally, increasing the color contrast of the inner chamber and the resolution of the camera can be helpful to accurately extract boundary further. The results of experiments verify that the extracted feature images are more effective than original images to recognize. For further study, we will continue to expand perceptual functions like tactile and sliding perception for more manipulation tasks.

Acknowledgements

This work was jointly supported by Tsinghua University Initiative Scientific Research Program No. 2019Z08QCX15, National Natural Science Foundation of China with Grant No. 91848206, and Foshan-Tsinghua industry–university-research cooperation collaborative innovation special fund No. 2018THFS04.

References

1. S. Furukawa, S. Wakimoto, T. Kanda and H. Hagihara, “A soft master-slave robot mimicking octopus arm structure using thin artificial muscles and wire encoders,” *Actuators.* **8**(2), 40 (2019).
2. Y. Zhang, Y. Liu and X. Sui, “A mechatronics-embedded pneumatic soft modular robot powered via single air tube,” *Appl. Sci.* **9**(11), 2260 (2019).
3. J. Nassour, V. Ghadiya, V. Hugel and F. H. Hamker, “Design of New Sensory Soft Hand: Combining Air-Pump Actuation with Superimposed Curvature and Pressure Sensors,” *Proceedings of the IEEE International Conference on Soft Robotics* (2018) pp. 164–169.

4. B. Fang, F. Sun, Y. Chen, C. Zhu, Z. Xia and Y. Yang, "A Tendon-Driven Dexterous Hand Design with Tactile Sensor Array for Grasping and Manipulation," *Proceedings of the IEEE International Conference on Robotics and Biomimetics* (2019) pp. 203–210.
5. H. Zhao, K. O'Brien, S. Li and R. F. Shepherd, "Optoelectronically innervated soft prosthetic hand via stretchable optical waveguides," *Sci. Rob.* **1**(1), 7529 (2016).
6. S. Ozel, E. H. Skorina, M. Luo, W. Tao, F. Chen, Y. Pan and C. D. Onal, "A Composite Soft Bending Actuation Module with Integrated Curvature Sensing," *Proceedings of the IEEE International Conference on Robotics and Automation* (2016) pp. 4963–4968.
7. B. Fang, H. Xue, F. Sun and H. Liu, "A cross-modal tactile sensor design for measuring robotic grasping forces," *Ind. Rob.* **46**(3), 37–44 (2019).
8. F. Sun, B. Fang, H. Xue and H. Liu, "A novel multi-modal tactile sensor design using thermochromic material," *SCIENTIA SINICA Inf.* **48**(48), 449 (2018).
9. C. Sferrazzać and R. D. Andrea, "Design, motivation and evaluation of a full-resolution optical tactile sensor," *Sensors* **19**(4), 928 (2019).
10. B. Fang, F. Sun, C. Yang, H. Xue, W. Chen, C. Zhang, D. Guo and H. Liu, "A Dual-Modal Vision-Based Tactile Sensor for Robotic Hand Grasping," *Proceedings of the IEEE International Conference on Robotics and Automation* (2018) pp. 1–9.
11. K. Aquilina, D. A. Barton and N. F. Lepora, "Principal Components of Touch," *Proceedings of the IEEE International Conference on Robotics and Automation* (2018) pp. 1–8.
12. J. Li, S. Dong and E. Adelson, "Slip Detection with Combined Tactile and Visual Information," *Proceedings of the IEEE International Conference on Robotics and Automation* (2018) pp. 7772–7777.
13. W. B. McInroe, L. C. Chen, Y. K. Goldberg, R. Bajcsy and S. R. Fearing, "Towards a Soft Fingertip with Integrated Sensing and Actuation," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (2018) pp. 6437–6444.
14. W. Yuan, Y. Mo, S. Wang and E. Adelson, "Active Clothing Material Perception Using Tactile Sensing and Deep Learning," *Proceedings of the IEEE International Conference on Robotics and Automation* (2018) pp. 1–8.
15. S. Luo, W. Yuan, E. Adelson, G. A. Cohn and R. Fuentes, "ViTac: Feature Sharing Between Vision and Tactile Sensing for Cloth Texture Recognition," *Proceedings of the IEEE International Conference on Robotics and Automation* (2018) pp. 2722–2727.
16. N. Pestell, J. Lloyd, J. Rossiter and F. N. Lepora, "Dual-modal tactile perception and exploration," *IEEE Rob. Autom. Lett.* **3**(2), 1033–1040 (2018).
17. B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M. E. Giannaccini, J. Rossiter and F. N. Lepora, "The TacTip family: Soft optical tactile sensors with 3D-printed biomimetic morphologies," *Soft Rob.* **5**(2), 216–227 (2018).
18. C. Xiang, J. Guo and J. Rossiter, "Soft-smart robotic end effectors with sensing, actuation, and gripping capabilities," *Smart Mater. Struct.* **28**(5), 055034 (2019).
19. A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Proceedings of the Conference on Advances in Neural Information Processing Systems* (2012) pp. 1097–1105.
20. Y. LeCun, K. Kavukcuoglu and C. Farabet, "Convolutional Networks and Applications in Vision," *Proceedings of IEEE International Symposium on Circuits and Systems* (2010) pp. 253–256.
21. A. F. Agarap, "Deep learning using rectified linear units (ReLU)," *arXiv preprint* arXiv.1803.08375 (2018).
22. P. Ramachandran, B. Zoph and Q. V. Le, "Searching for activation functions," *arXiv preprint* arXiv.1710.05941 (2017).
23. C. Gulcehre, M. Moczulski, M. Denil and Y. Bengio, "Noisy Activation Functions," *Proceedings of International Conference on Machine Learning* (2016) pp. 3059–3068.
24. X. Qi, T. Wang and J. Liu, "Comparison of Support Vector Machine and Softmax Classifiers in Computer Vision," *Proceedings of International Conference on Mechanical, Control and Computer Engineering* (2017) pp. 151–155.
25. D. Sculley, "Combined Regression and Ranking," *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2010) pp. 979–988.
26. S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint* arXiv.1069.04747 (2016).
27. O. Shamirć and T. Zhang, "Stochastic Gradient Descent for Non-smooth Optimization: Convergence Results and Optimal Averaging Schemes," *Proceedings of International Conference on Machine Learning* (2013) pp. 71–79.
28. N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.* **12**(1), 145–151 (1999).
29. B. O'donoghueć and E. Candes, "Adaptive restart for accelerated gradient schemes," *Found. Comput. Math.* **15**(3), 715–732 (2015).
30. A. C. Wilson, R. Roelofs, M. Stern, N. Srebro and B. Recht, "The Marginal Value of Adaptive Gradient Methods in Machine Learning," *Proceedings of the Conference on Advances in Neural Information Processing Systems* (2017) pp. 4148–4158.
31. M. D. Zeiler, "ADADELTA: An adaptive learning rate method," *arXiv preprint* arXiv.1212.5701 (2012).

https://doi.org/10.1017/S0263574720000429 Published online by Cambridge University Press

32. F. Zou, L. Shen, Z. Jie, W. Zhang and W. Liu, “A Sufficient Condition for Convergences of Adam and RMSProp,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019) pp. 11127–11135.
33. D. P Kingmać and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint* arXiv.1412.6980 (2014).
34. A. R. Forrest, “Interactive interpolation and approximation by Bẽzier polynomials,” *Comput. J.* **15**(1), 71–79 (1972).

https://doi.org/10.1017/S0263574720000429 Published online by Cambridge University Press