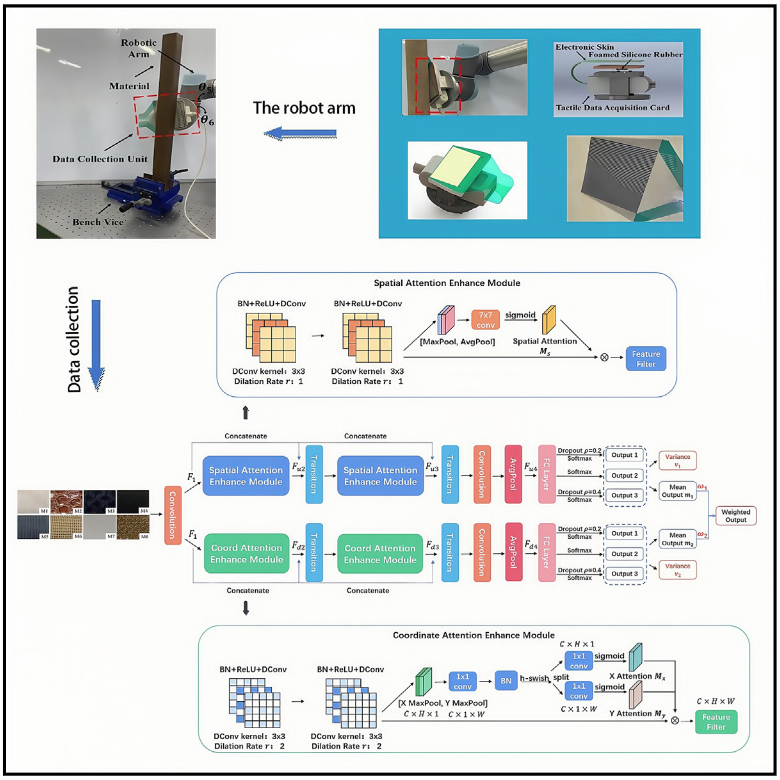
iScience

Article

A deep neural network for tactile perception in open scenes

Graphical abstract



Highlights

- Three batches of tactile datasets are collected to simulate the open scenes
- A multi-receptive field attention enhancement neural network is proposed for the open scenes
- The spatial attention and coordinate attention enhancement modules are used in our model

Authors

Huirong Fang, Qianhui Yang, Kunhong Liu, Xiangyi Huang, Yu Xie

Correspondence

lkhqz@xmu.edu.cn (K.L.),
xieyu@xmu.edu.cn (Y.X.)

In brief

Kinesiology; Robotics

Fang et al., 2025, iScience *28*, 112330
May 16, 2025 © 2025 The Author(s). Published by Elsevier Inc.
https://doi.org/10.1016/j.isci.2025.112330

CellPress

iScience

CellPress
OPEN ACCESS

Article

A deep neural network for tactile perception in open scenes

Huirong Fang,[1,2,6] Qianhui Yang,[3,6] Kunhong Liu,[3,4,7,*] Xiangyi Huang,[5] and Yu Xie[5,*]

[1]School of Electronic Information, Zhangzhou Institute of Technology, Zhangzhou 363000, China
[2]Intelligent Monitoring of the Fujian Provincial Higher Education Application Technology Engineering Center, Zhangzhou 363000, China
[3]Digital Media Technology Department, Film School of Xiamen University, Xiamen 361102, China
[4]Key laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan, Ministry of Culture and Tourism, Xiamen 361102, China
[5]Pen-Tung Sah Institute of Micro-Nano Science and Technology, Xiamen University, Xiamen 361102, China
[6]These authors contributed equally
[7]Lead contact
*Correspondence: lkhqz@xmu.edu.cn (K.L.), xieyu@xmu.edu.cn (Y.X.)

https://doi.org/10.1016/j.isci.2025.112330

SUMMARY

Tactile perception is important for the robots to understand their working environment. While in real-world applications, robots usually must face unexpected changes in external conditions, such as the re-installation of the robot end effector or the change of the installation location. Consequently, the collected tactile material data tend to vary to a certain extent, which brings great difficulties to the tactile perception. To handle this problem, different from the former studies of tactile perception in enclosed environments, this study focuses on the tactile material recognition task using robot electronic skin in open scenes. We construct a cross-batch tactile dataset to simulate open scenes and propose the multi-receptive field attention enhancement network (MRFE) to handle tactile material recognition. Compared with other machine learning algorithms, experiments show that the proposed method overcomes the problem of data drift caused by changes in posture, contact force, sliding velocities, exploratory motions, and assembly conditions.

INTRODUCTION

The sense of touch is one of the essential ways of human interaction. The tactile sensors provide robots the ability to perceive the real environment, stay away from potentially destructive effects, and acquire information for subsequent tasks such as hand manipulation.[1] From the 1980s, researchers began to design tactile sensors and proposed measurements based on force-sensing resister, capacitive, piezoelectric, and other working principles.[2] Since electronic skin has excellent characteristics of softness, ease of adhesion, and humanoid feeling, it has become the mainline of today's tactile sensing equipment.[3]

In recent years, tactile sensors have promoted further exploration of material recognition,[4] grasping,[5] pressure and temperature sensing,[6] and other aspects.[7] Among these applications, material recognition is a mandatory capability for some types of robot systems, such as service robots, medical robots, and exploratory robot systems.[8] However, the safe and efficient operation of these robots in unstructured environments remains to be a key research area. The difficulty of this task lies in the challenges caused by the complexity of the environment, the hard interpretation of sensor data, and the uncertainties in the integration of various systems. Thus, there arises an urgent need for precise and stable tactile perception to deal with open scenes that may change at any time. In previous works on tactile material recognition, many studies attempted to build tactile perception frameworks based on diverse machine learning algorithms[9–13] and achieve excellent performance. However, most of the relevant research work was carried out in enclosed environments,[14] which means strictly controlled experimental settings with fixed speeds and forces. An example is that in Ref.[10], researchers used a constant speed to move over the materials. To the best of our knowledge, there are only two papers considering the impacts of the disassembly and assembly of the experimental devices. In Ref.[9], authors tried sensor relocation offsets to construct datasets at different levels of difficulty. And their experiments showed that the sensor relocation offsets caused a sharp decrease in the tactile material recognition performance of the proposed model. Another work by Chen et al.[15] also explained that the introduction of tactile elastomeric substrate would bring uncertainty to tactile data. Furthermore, Liu et al. used generative adversarial networks (GANs) to synthesize open-set samples as unknowns in Ref.[16], to better handle the open space risk.

To solve the tactile inconsistencies problem, an intuitive way is to consider the robot kinematics for calibration operations, which includes calibration of robots and sensors. The robot calibration technique makes it relatively easy to obtain higher positioning accuracy. But for the array tactile sensor, the measurement error is about ±10%, and there are common problems

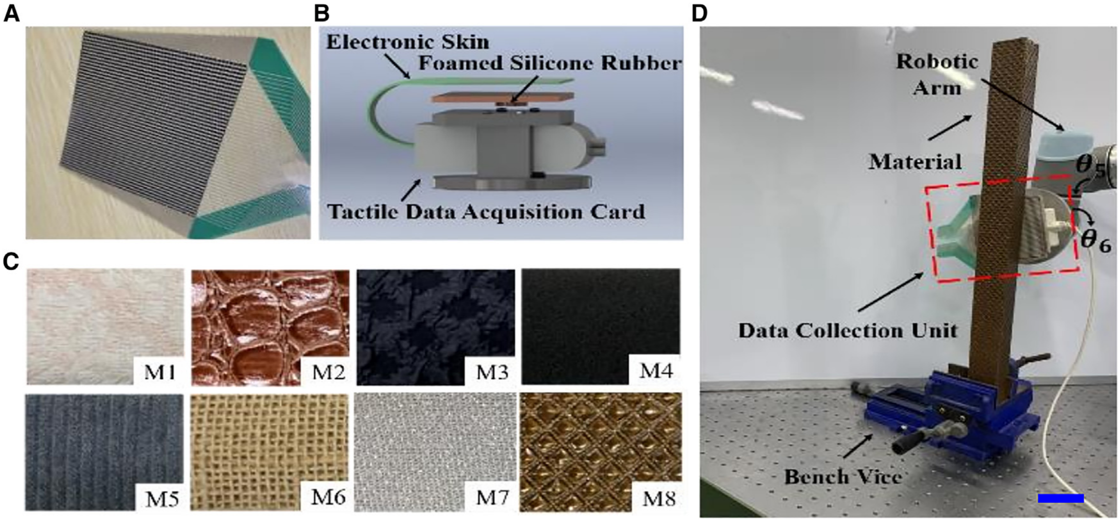


Figure 1. Experimental settings

(A) The tactile sensor.

(B) The tactile data collection unit.

(C) Eight types of test materials: M1 *coarse towel*, M2 *crocodile pattern*, M3 *relief cloth*, M4 *sponge*, M5 *horizontal fabric*, M6 *linen*, M7 *gauze*, and M8 *diamond pattern*.

(D) The tactile data collection unit installed on the UR5 robotic arm collected data (the length bar is 5 *cm*).

such as creep and reuse errors.[17,18] What makes it worse is that there is no standard calibration method for tactile sensors mounted on robots so far.

This study tries to extend the tactile perception problem from the enclosed scenes to the open scenes, serving as the first attempt at the challenges of the material tactile perception under changed conditions. This study simulates the situations of identifying tactile material using robot arms in open scenes, and the 44 × 44 array of tactile data is obtained with the electronic skin fixed on the robotic arm, touching seven types of fabrics with changing positions and forces. Different from the traditional experiment settings, our experiments are conducted under various initial conditions to collect different batches of data, such as re-disassembling the electronic skin or the end effector and changing the exploratory motions. This can simulate the actual working scenario that the robot's tactile perception would face in the real world. Data analysis and further experiments show that the features of different types of fabrics fluctuate greatly across different batches of tactile data, and it is difficult to directly handle the cross-batch tactile data by using conventional neural network models, such as ShuffleNet and ResNet, in the training of a single enclosed environment.

This study proposes the multi-receptive field attention enhancement network (MRFE) to explore a robust tactile perception model for open scenes. By increasing the diversity of attention areas through different receptive fields and different attention modules from two paths, an attention enhancement mechanism is designed to strengthen important feature subsets, and the uncertainty of each path's output is quantified so that the prediction from two paths is reweighted and fused based on corresponding uncertainty.

In summary, the main contributions of this paper include the following:

(1) A cross-batch tactile dataset under the change of initial experiment conditions to simulate open scenes. This breaks existing assumptions of the enclosed experimental settings limited to the same strict initial experimental conditions.
(2) A deep-neural-network-based solution for accurate material recognition in open scenes, i.e., under various unknown conditions (different poses, contact forces, sliding velocities, exploratory motions, and assembly conditions).

Related work

The interpretation of tactile sensor readings is closely bound up with the sensors used.[19] Most research in tactile material recognition was carried out using multi-modal tactile sensors or high spatial resolution tactile arrays analogous to human fingertips for tactile sensing.[20] For the original tactile sensor readings, some researchers designed tactile descriptors to carry out feature expression and then applied machine learning algorithms to recognize diverse materials.[21]

Fishel et al.[22] used a testbed equipped with the BioTac, a highly sensitive multimodal tactile sensor, for exploring 117 textures by sliding movements. Based on some related experiments, they chose three combinations of force and speed as the best exploratory movements. That is *1.26 N* and *1 cm/s* for discrimination based on traction, *0.2 N* and *6.31 cm/s* for discrimination based on roughness, and *0.5 N* and *2.5 cm/s* for discrimination based on fineness. Based on the features of traction from motor

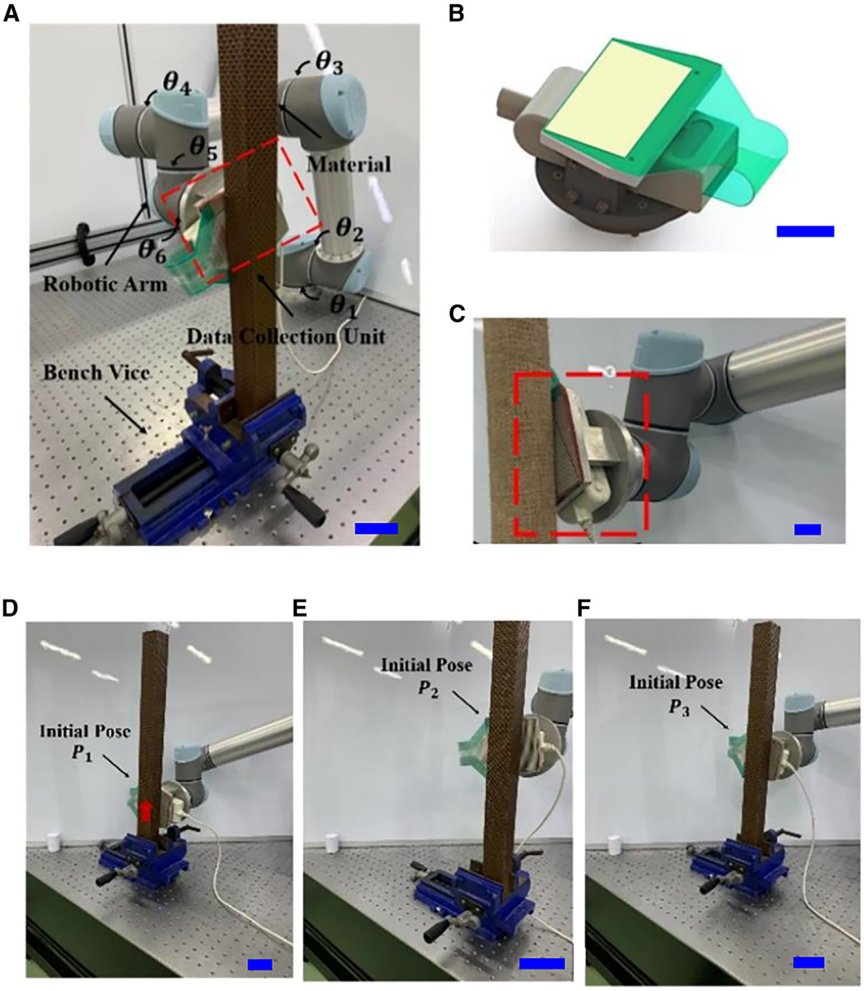


Figure 2. The illustration of experiment settings

(A) The robot arm with 6° of freedom $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$.

(B) The tactile data collection unit, containing the e-skin on the double-layer foamed silicone rubber.

(C) The installation of the tactile data collection unit on the robot arm.

(D) Sliding exploration: using the robot arm sliding upward from the bottom (P_1) with the fixed force to collect *Batch* 1 data; pressing exploration: using a new initial pose (P_2), and the motions change from slide to press with different forces, to collect *Batch* 2 data.

(E) Pressing exploration: using an random initial pose (P_3), with other setting keeping the same as those for *Batch* 2 to collect *Batch* 3 data. The length bar is 5 *cm* in all sub-figures.

current, roughness, and fineness, the method obtained an accuracy of 95.4%. However, the sample data were sampled under the precisely controlled testbed, and another work[23] showed that the performance declined dramatically when transferred to a real robotic setup.

Kaboli et al.[13] designed a set of basic tactile descriptors to represent the statistical properties of tactile signals in time domains, activity, mobility, and complexity. In the experiment setup, they deployed BioTac on the fingertips of the Shadow Hand sliding on the material for data collection. The maximum contact force was strictly controlled at *3 N*, and the maximum sliding speed was *4 cm/s*. Based on the designed tactile descriptors, the proposed SVM successfully classified 120 materials with an accuracy of 100%. Results showed that the proposed tactile descriptors were robust enough to ignore the distinction of tactile sensing technology. However, such tactile descriptors were invariant only for a particular exploration movement and the corresponding parameters.

Recently, much research began to deploy deep learning methods to automatically learn the self-organizing features of original tactile sensor readings. Baishya et al. were the first to apply deep learning to tactile material recognition.[9] In their experiment, the Tekscan 4256E sensor patch was connected to the robot's left thumb and slid on the tube material at a constant speed of *3 cm/s*, with the contact force controlled at about *1 N*. It is noteworthy that three datasets of different classification difficulty levels were constructed: (1) T-LAB recorded once through sensor connection without repositioning; (2) T-REAL removed and reconnected the sensor several times with as high a repositioning accuracy as possible; (3) T-HARD repositioned the sensor again, but with an offset about 4 mm and a more significant spatiotemporal signal change. When the T-REAL-based model was transferred to T-HARD for testing, the accuracy obtained by other classifiers would decrease to 70%, whereas the deployed CNN could reach 91.7%, showing certain robustness to sensor relocation errors. Besides, Fang et al. proposed a CNN integrated with attention mechanism to accomplish the defect detection task, with the frequency domain filtering weakening the influence of fabric texture information.[24]

Researchers[10] proposed a fast texture classification framework by utilizing a spiking neural network (SNN) to learn from the neural coding of the conventional tactile sensor readings. The SynTouch BioTac sensor[25] was used to move at a constant speed of *2.5 cm/s* over a linear trajectory of 20 cm, reaching the best speed given for texture classification[22] with a limited force range of *0–2 N*. With this setup, 20 different types of materials were collected with 50 samples in each type. The authors also used the iCub RoboSkin tactile sensor[26] to collect data without strict force and speed control. The accuracy score of the proposed framework in the BioTac dataset was about 94%, whereas the accuracy score of the RoboSkin dataset was 92.2%, lower than that of LSTM but much higher than that of SVM. The authors considered that the reason that lay in BioTac's data collection setup was

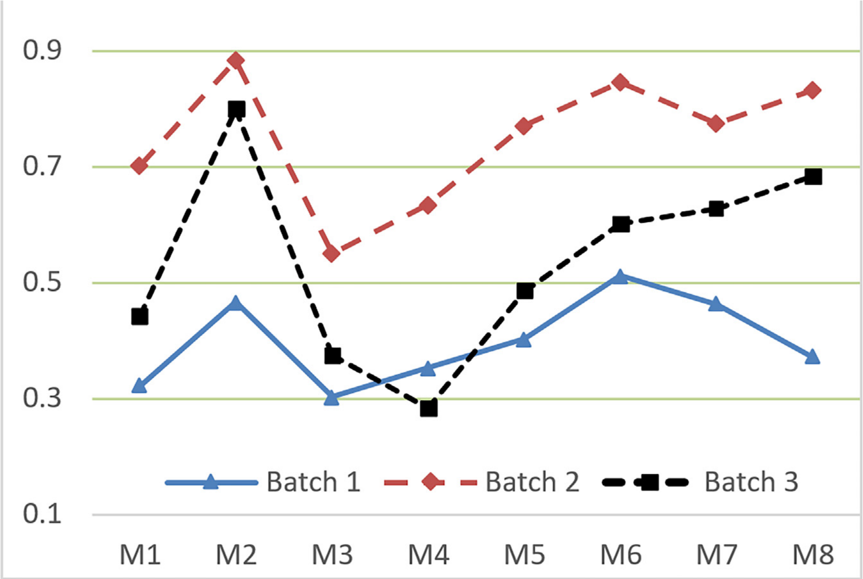


Figure 3. The similarity of three batches by using *Batch* 1 data as the benchmark

M1–M8 represent 8 materials: *coarse towel*, *crocodile pattern*, *relief cloth*, *sponge*, *horizontal fabric*, *linen*, *gauze*, *and diamond pattern*.

much more rigorous than that of RoboSkin's and thus provided a very clean dataset.

These studies all employed a single exploratory motion, such as touch or slide over materials for tactile sensing. A recent study[27] considered a combination of sliding and touch movements for tactile recognition. In this study,[27] Taunyazov et al. used the iCub robot to explore the materials fixed on a non-deformable metal surface, and taxels on the iCub forearm can obtain tactile information. For touch movements, the robotic shoulder joint angle was altered from *87*° to *93*° with angular

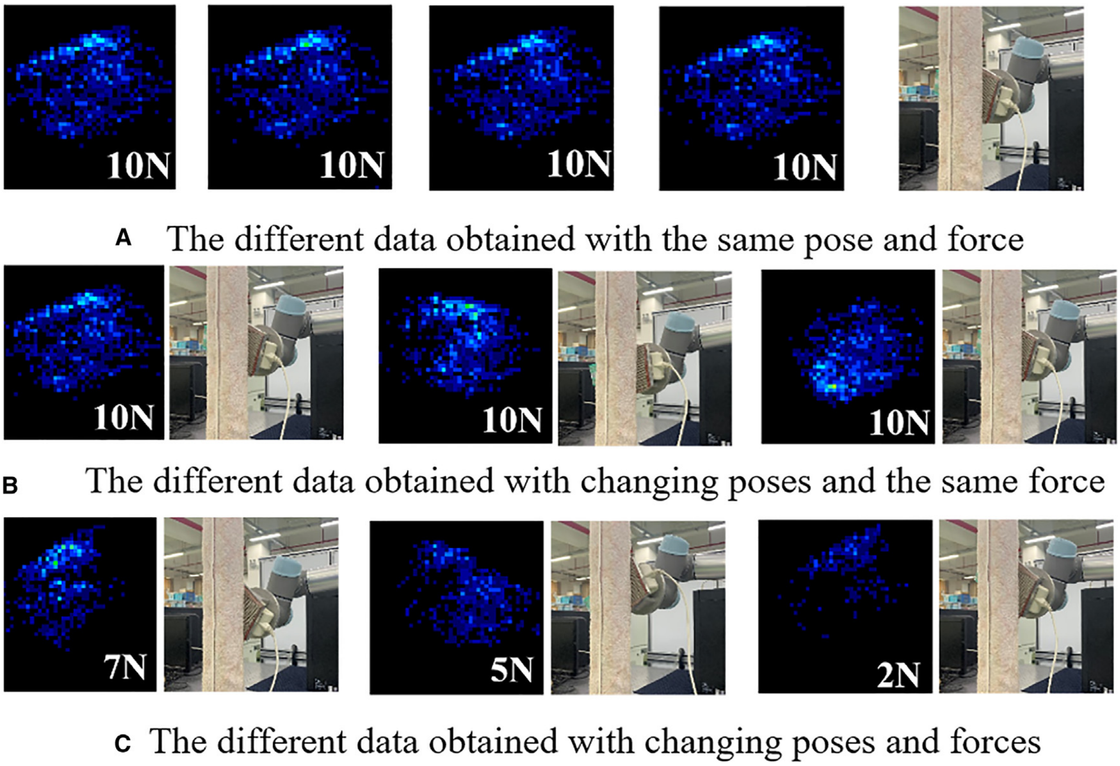

10N
10N
10N
10N
A The different data obtained with the same pose and force
10N
10N
10N
B The different data obtained with changing poses and the same force
7N
5N
2N
C The different data obtained with changing poses and forces

Figure 4. Some examples of collected data in the *coarse towel* class obtained under different conditions

Table 1. The Dunn indices of different class pairs

	M1	M2	M3	M4	M5	M6	M7	M8
M1	–	4.65	2.23	2.34	3.58	3.76	2.84	5.43
M2	–	–	4.15	3.26	3.67	8.12	5.17	9.25
M3	–	–	–	2.25	2.05	5.10	3.08	3.73
M4	–	–	–	–	1.76	3.18	2.20	3.26
M5	–	–	–	–	–	3.05	2.81	4.43
M6	–	–	–	–	–	–	4.85	8.15
M7	–	–	–	–	–	–	–	5.08

velocity $1^{\circ}/s$, whereas the elbow joint angle changed from 90° to 30° with angular velocity $5^{\circ}/s$ for sliding movements. Three ML models, SVM, SVM-LSTM, and CNN-LSTM, were trained with touch, sliding, and a combination of touch and sliding data. And the results showed that different texture characteristics were obtained during touch and slide, and the texture recognition performance of models can be improved by combining touch and slide datasets for training. Although joint data training is a proper solution, it does not take the impact of different movement ways, such as touching and sliding, into consideration. For example, when the model embedded in a robot is trained on a single slide dataset, but the robot needs to explore the environment with a single quick touch, this type of tactile perception across exploratory motions remains a challenging problem.

In most cases, the state-of-the-art approaches are executed under precisely controlled experimental settings with the constant velocity, force, and a single exploratory motion.[28] Some researchers[9,22] reported that the recognition results would deteriorate when the models trained under the controlled laboratory settings were applied to more realistic robotic setups. In their work,[29] Yang et al. used a transfer learning model to handle the changed data in different experimental conditions. Although, there is still no further explorations for this problem. And in this study, we try to tackle the open scenes with a deep neural network model that fits various changing conditions, such as different poses, contact forces, sliding velocities, exploratory motions, and assembly conditions.

Experiment settings

The setup of the robot arm

We use electronic skin (Pressure Mapping Sensor 5076, Tekscan, USA) to obtain tactile data, as shown in Figure 1A. In Figure 1B, a tactile data collection unit is designed to attach the sensor to the end of the robotic arm. The sensor is characterized by light weight, thin thickness, flexible, and easy conformation. To enhance the imaging quality when the e-skin is in contact with the material,[30] a double-layer foamed silicone rubber is added under the e-skin. The first layer is close to the e-skin, the shape is consistent with the e-skin, and the thickness is 5 mm. The second layer is a cylinder with the diameter of 25 mm and the thickness of 5 mm. The e-skin and double-layer foamed silicone rubber are pressed on the plane plate of I-shaped structure to form a convex structure. The array sensor has 44 × 44 sensing units, a spatial resolution of 27.6 cells/cm^2, and a pressure range of 50 PSI. The e-skin can be set to obtain data at the fastest rate of 10 ms/frame, and obtained data are in the form of a list with 1936 rows. The data are reorganized and converted into a 44 × 44 matrix, representing a tactile image.

Table 2. Texture classification accuracy scores

Models	*2 cm/s_val*	*4 cm/s*	*6 cm/s*	*8 cm/s*
ShuffleNet	99.93 ± 0.04	75.21	66.71	56.06
ResNet	100.00 ± 0.00	80.83	67.25	64.35
DenseNet	99.76 ± 0.00	88.00	81.60	84.71
GCPL	99.82 ± 0.02	86.35	79.27	80.42
STAM	99.97 ± 0.01	85.42	82.43	79.64
MRFE(ours)	100.0 ± 0.00	90.25	84.63	83.25

At the same time, a data acquisition card (DAQ)[31] is used to collect data with a high acquisition frequency of 100 Hz. The data acquisition card is installed in the reserved space of I-shaped structure. The tactile data collection unit is installed on a 6-DOF robotic arm (UR5, Universal Robots, Denmark), which is characterized by its lightness. It can perform repetitive tasks with a load of up to 5 kg and a repeatability of ±0.03 mm.

Eight types of square sticks made by different materials with the same shape are chosen, as shown in Figure 1C. The different types of materials are selected under the consideration that the selected material should contain intact and regular texture patterns, so that it can support stable tactile data.

For exploration, different sticks are clamped by a bench vice in turn, as shown in Figure 1D. The data collection unit is installed in the robot arm, which has 6° of freedom. In this way, the robot arm moves and touches the stick to generate the tactile data, which is collected for tactile material recognition.

Cross-batch data collection and analysis

To simulate the data diversity brought by open scenes, we change the poses of the robotic arm before obtaining each batch of data. First, θ_5 and θ_6 are changed in the joint angle control parameter θ $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ of the robotic arm (see Figure 2A), which are the closest to the data collection unit and have the greatest impact on tactile data imaging. And then z is changed by the pose P (x, y, z, R_x, R_y, R_z) of the end effector of the robotic arm. After adjusting the parameters, the data are collected again by pressing various materials. Through the above two steps, the style of obtaining sample data is greatly changed. The force reflected by the e-skin is the sum of the force values of all the sensing points:

$$F(t_k) = \begin{bmatrix} f_{0,0}(t_k)\cdots & f_{0,M}(t_k) \\ \vdots \ddots & \vdots \\ f_{N,0}(t_k)\cdots & f_{N,M}(t_k) \end{bmatrix} \qquad \text{(Equation 1)}$$

$f_{i,j}(t_k)$ is the force value reflected by the sensor unit in row i and column j at time t_k. N and M represent the rows and columns of the e-skin. To simplify the data collection process, the resultant force value embodied by the e-skin is used as its interaction force in contact with the material.

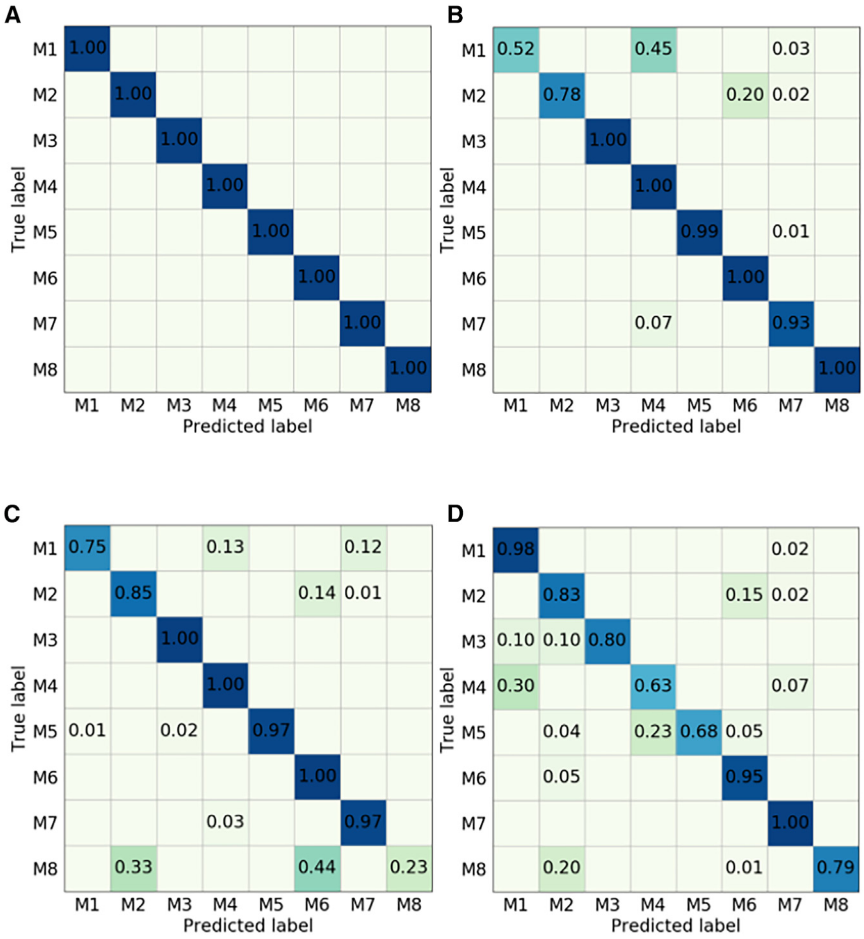


Figure 5. The confusion matrices obtained by MRFE trained on a dataset sampled at 2 cm/s and tested on datasets with different velocities

(A) 2 cm/s.
(B) 4 cm/s.
(C) 6 cm/s.
(D) 8 cm/s.

We collect three batches of tactile data based on the tactile sensor, and the data collection process is given as follows:

(1) *Batch 1*: the robotic arm initially presses the bottom area of the material with nearly *10 N* force and then slides upward from the bottom with touching the material for 2 s. Every sample includes 200 frames. To avoid duplication of information, a frame of tactile image is taken every 50 frames. In this way, we get 150 samples at each kind of arm movement speed, and there are four kinds of movement speeds for data collection: *2 cm/s*, *4 cm/s*, *6 cm/s*, and *8 cm/s.* The experimental platform is shown in Figure 2A.

Table 3. Texture classification accuracy scores

Models	*2+4+6 cm/s_val*	*8 cm/s*
ShuffleNet	98.19 ± 0.33	88.04
ResNet	99.98 ± 0.01	96.94
DenseNet	98.86 ± 0.00	97.71
GCPL	100.00 ± 0.00	96.32
STAM	99.97 ± 0.01	95.26
MRFE(ours)	100.00 ± 0.00	97.75

(2) *Batch 2*: remove and reinstall the sensor, with fixing the table and material used in *Batch* 1. The initial pose of the robotic arm is changed, and the exploratory motions of sampling change from slide to press compared with that of *Batch* 1. After pressing the materials, the tactile data are recorded as 1 frame. Four kinds of forces are deployed for data collection respectively: *2 N*, *5 N*, *7 N*, and *10 N.* And 150 samples are gathered for each force, as shown in Figure 2B.

(3) *Batch 3*: remove and reinstall the sensor, with fixing table and material used in *Batch* 2. The initial pose of the robotic arm is also changed compared with that of *Batch* 1 and *Batch 2.* The sampling strategy is the same as *Batch* 2, but 450 samples are recorded for each force, as shown in Figure 2C.

The SSIM[32] measurement is used to represent the average structural similarity index between two images. The larger SSIM index indicates the higher similarity between the two images. Its definition is given as follows:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{\left(\mu_x^2+\mu_y^2+C_1\right)\left(\sigma_x^2+\sigma_y^2+C_2\right)} \quad \text{(Equation 2)}$$

μ_x and μ_y represent the mean values of images x and y, respectively, whereas σ_x^2 and σ_y^2 represent the variances of images x and y. σ_{xy} is the covariance of images x and y. $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ are two constants. Generally, $k_1 = 0.01$, $k_2 = 0.03$, and $L = 255$.

The SSIM measurement is deployed to evaluate the similarity among different batches. First, we select the benchmark images of each category in *Batch* 1 data and then calculate the similarity of samples in each category in *Batch* 1. Then we calculate the similarity of *Batch* 2 and *Batch* 3 data, respectively, by the comparisons with the benchmark images of *Batch* 1. For example, given a rough towel image x of *Batch* 1 data as the benchmark, y_i, m_i, and n_i represent the images of rough towel of *Batch* 1 (x excepted), *Batch* 2, and *Batch* 3 data, respectively. $SSIM(x,y_i)$ is calculated with all remaining rough towel images y_i from *Batch* 1 data, and the mean value of $SSIM(x,y_i)$ is taken as the internal

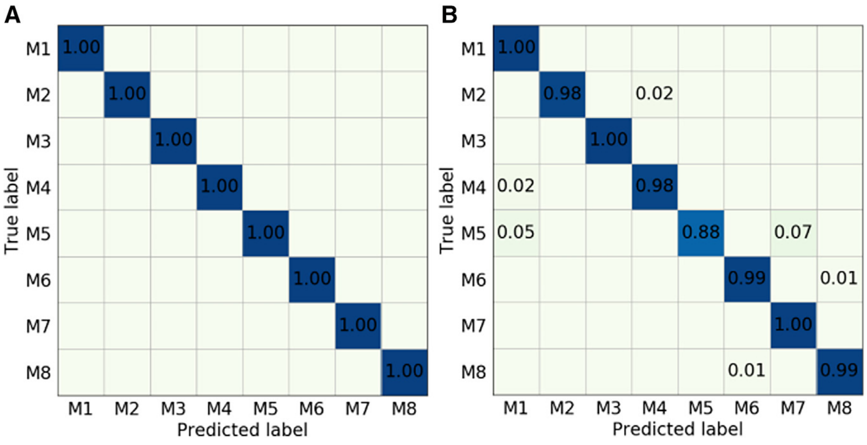


Figure 6. The confusion matrices obtained by MRFE with being trained on the joint dataset sampled at the velocity of *2 cm/s*, *4 cm/s*, and *6 cm/s*

(A) The test results on the joint dataset with three types of velocities; (B) The test results on the dataset sampled at *8 cm/s.*

similarity of rough towels from *Batch* 1. Similarly, a benchmark image x is compared one by one with images m_i and n_i from the other two batches to calculate the mean $SSIM(x, m_i)$ and the mean $SSIM(x, n_i)$, as the cross-batch similarity based on the *Batch* 1 data.

As shown in Figure 3, the vertical axis represents image similarity. The *Batch* 1 data have four highly similar categories (M2, M5, M6, M7, and M8), with a similarity greater than 0.7. In contrast, the similarity of *Batch* 2 and *Batch* 3 data to the benchmark image decreased significantly, and some categories' similarity is even lower than 0.3. This indicates that the distribution of features varies greatly among different batches in the same category. In the comparison of the cross-batch images, the decreases of similarity of M1, M3, and M4 reach the highest levels, indicating that these classes are more difficult in the cross-batch recognition task than others.

Figure 4 shows some examples of the collected *coarse towel* data on different batches of data. It is found that the data collected under the same conditions (the same force and pose) tend to be very similar (Figure 4A). Therefore, the data in the close environment are easy to be identified. However, with different poses or forces, the data collected in different batches would be quite different (Figures 4B and 4C). And the higher difference of conditions leads to higher diversity in the data of the same category. Consequently, the identification of such diverse data would become a hard problem. And the higher difference in the settings leads to higher diversity in data.

Moreover, the small margins among different classes means that the identification of different classes would be difficult and complex. To better quantitatively analyze the overlapping of different classes, the *Dunn index*[33] is adopted to evaluate the intra-class distance by

$$D = \frac{\min\limits_{1 \le k \le k' \le m} d_{min}(C_k, C_{k'})}{\max\limits_{1 \le l' \le m} diam(C_{l'})} \quad \text{(Equation 3)}$$

where m is the number of clusters, and $d_{min}(C_k, C_{k'})$ measures the degree of dispersion between any two clusters C_k and $C_{k'}$. $diam(C_{l'})$ measures the intra-class distance within class $C_{l'}$. Therefore, a smaller Dunn index means the harder classification task.

The Dunn indices between every class pair of *Batch 1* are given in Table 1, showing that the Dunn indices of different class pairs vary greatly. It indicates that the data of different classes vary across different classes in the training set. Although data of M2 are quite different from that of M8, the data of most of classes are close to each other. So they are easy to be misclassified to each other, indicating that the discernibility of tactile data of different categories in the same batch is poor, and some categories have similar data distributions. The reason lies in that the similarity between the texture features of the two types of materials reveals the similar shapes of the corresponding tactile data.

Table 4. Texture classification accuracy scores

Models	*2+5 N_val*	*6 N*	*7 N*	*8 N*	*9 N*	*10 N*
ShuffleNet	99.25 ± 0.52	96.06	91.03	86.44	68.06	54.13
ResNet	99.98 ± 0.01	99.83	99.14	98.44	94.36	90.75
DenseNet	99.05 ± 0.01	99.72	99.06	96.33	92.92	92.63
GCPL	99.52 ± 0.01	99.81	99.25	97.32	93.26	93.44
STAM	99.63 ± 0.01	99.77	99.42	98.25	94.13	94.36
MRFE(ours)	99.81 ± 0.09	99.92	99.47	98.14	95.75	95.50

RESULTS AND DISCUSSION

In this section, we present the experimental results of the proposed model and other comparable models (ShuffleNet,[34] ResNet,[35] DenseNet,[36] GCPL,[37] and STAM[38]) on three datasets with different degrees of openness. In the following sections, we discuss the influence of pressing force and sliding speed on material recognition and the generalization ability of models, respectively. Further, we also explore material recognition across batches of data that are closer to the real scene and verify the robustness of the proposed model.

The cross-velocity experiments

The cross-velocity experiments are conducted on the *Batch* 1 data that are sampled at four sliding speeds over materials for 2 s: *2 cm/s*, *4 cm/s*, *6 cm/s*, and *8 cm/s*. The first group of experiments is obtained by training on a dataset sampled at the low sliding velocity of *2 cm/s* and testing on the remaining three datasets with higher sampling velocity. The second group of experiments is conducted by training on the joint dataset sampled at several sliding velocities of *2 cm/s*, *4 cm/s*, and *6 cm/s* and testing on the dataset with a sampling velocity of *8 cm/s*. Each training data is split into the training and validation sets at 2:1 proportion.

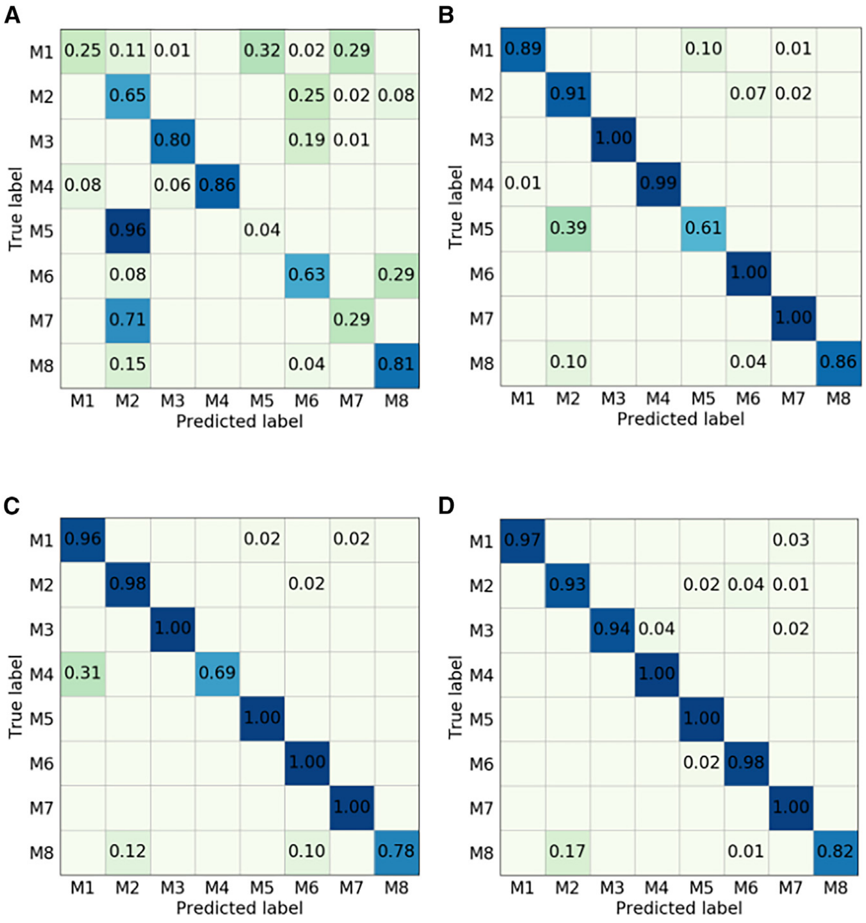


Figure 7. The confusion matrices obtained by models trained on the joint dataset sampled at the forces of *2 N* and *5 N*

(A) The confusion matrix obtained by ShuffleNet on the dataset with a *10 N* sampling force.

(B) The confusion matrix obtained by ResNet on the dataset with a *10 N* sampling force.

(C) The confusion matrix obtained by DenseNet on the dataset with a *10 N* sampling force.

(D) The confusion matrix obtained by MRFE on the dataset with a *10 N* sampling force.

It can be seen from Table 2 that when the models trained and tested on the dataset gathered at the same velocity, their accuracy scores are over 99%. But their performance cannot be maintained at the same level when the test data are obtained at other velocities. Due to the diversity of data obtained at different sampling velocities, it is difficult for a model trained on a dataset sampled at a single velocity to be directly used to predict the dataset collected at other high velocities. From Table 2, the accuracy of ShuffleNet and ResNet on the dataset sampled at the velocity of *8 cm/s* is only 56.06% and 64.35%, respectively. In contrast, DenseNet, GCPL, and STAM can reach better accuracy score of 84.71%, 80.42%, and 79.64%, generating higher performance on different velocities, whereas the proposed MRFE model achieves the highest accuracy score of 83.25%, confirming the effectiveness of our model.

Table 5. Texture classification accuracy scores

Models	*Batch* 1_val	*Batch* 2	*Batch* 3
ShuffleNet	99.30 ± 0.96	59.79	46.43
ResNet	100.00 ± 0.00	85.46	63.17
DenseNet	99.42 ± 0.01	80.35	66.83
GCPL	99.94 ± 0.01	86.81	72.73
STAM	99.96 ± 0.01	88.57	70.62
MRFE(ours)	100.00 ± 0.00	88.50	75.50

Figure 5 presents the recognition accuracy of each material category. From Figure 5B, in the prediction of the dataset sampled at the velocity of *4 cm/s*, the confusion between M1 (*coarse towel*), M4 (*sponge*), and M7 (*gauze*) is the main reason for the overall accuracy decline, and the recognition accuracy of M1 is only 52%, and 45% of M1 samples are misclassified to M4. This is caused by the similar roughness characteristics of *rough towel* and *sponge*, and this similarity leads to a higher difficulty level in the prediction of cross-velocity datasets. This also appears in the prediction of the dataset sampled at the velocity of *6 cm/s* and *8 cm/s* where the recognition accuracy of M1 is 75% (see Figure 5C) and M4 is 63% (see Figure 5D). In (Figure 5C), due to the similar periodic texture of M6 (*linen*) and M8(*diamond pattern*), 44% of *diamond pattern* are misclassified as *linen*. In Figure 5D, because M2 (*crocodile pattern*) and M3 (*relief cloth*) have irregular texture patterns, the tactile information obtained from different touch directions is very different, resulting in a slightly worse recognition accuracy of 83% and 80%, respectively, in the dataset sampled at *8 cm/s*.

In Table 3, it is found that compared with Table 2, the accuracy of prediction on the dataset sampled at the velocity of *8 cm/s* has been improved to a certain extent by combining multi-speed datasets for model training. The joint multi-speed datasets contain more diverse tactile information, and the dataset with similar sampling velocity may have similar features, which helps to improve the recognition accuracy of the dataset with higher sampling velocity. The confusion matrix obtained by MRFE trained on the joint dataset sampled gets 100% accuracy when tested at the joint dataset with three types of velocities, as shown in Figure 6A. The fusion of different data produces a surprisingly high result.

In this experiment, all models except for the ShuffleNet perform well, achieving close accuracy on the dataset collected at *8 cm/s* speed. Compared with the results in Table 2, the test results are all improved, and the reason may lie in the larger scale

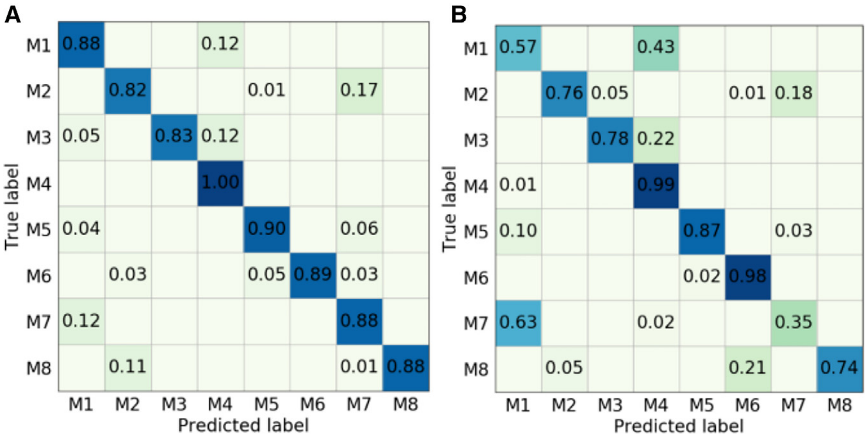


Figure 8. The confusion matrices by MRFE trained on *Batch* 1 data

(A) The confusion matrix on the *Batch* 2 data.

(B) The confusion matrix on the *Batch* 3 data.

of training data. Whereas it is found that our model can still generate the best performance in this case. Compared with the results in Figures 6B and 5D, the performance of the model trained on the joint datasets has been greatly improved in the prediction of M2 (*crocodile pattern*), M3 (*relief cloth*), M4 (*sponge*), and M8 (*diamond pattern*), with an accuracy score higher than 98%. It is because these textures are more concave and convex than others.

The cross-force experiments

The cross-force experiments are conducted on the *Batch* 3 data that are collected by pressing the materials with four forces: *2 N*, *5 N*, *7 N*, and *10 N*. To better observe the robustness of models on the variable forces, data collected under the force of *6 N*, *8 N*, and *9 N* are added to this experiment as test sets for evaluation. The experiment is conducted by training on the joint dataset sampled at two forces of *2 N* and *5 N* and testing on the dataset with other higher sampling forces of *6 N*, *7 N*, *8 N*, *9 N*, and *10 N*. The training data is also split into the training and validation sets at 2:1 proportion.

From Table 4, in the prediction of datasets collected under *6–10 N* forces, the accuracy of the model trained with low sampling force data to predict larger sampling force data shows a declining trend, dropping from 96.06% to 54.13% in ShuffleNet and 90.75% in ResNet in the material recognition accuracy of *10 N* sampling strength data. This may be due to the poor quality of the tactile image obtained by pressing the material with low force, with fuzzy and incomplete edge information. Whereas the remaining models promise better performance regardless of the change of forces, and in nearly all cases, our model achieves the best performance.

As can be seen from Figure 7, ShuffleNet and ResNet only have 4% and 61% recognition accuracy of M5 (*horizontal fabric*), and most samples are misclassified into M2 (*crocodile pattern*). We find that the *crocodile pattern* has an irregular shape, and some tactile image samples obtained by pressing it with a small force only contain the local texture of *crocodile pattern*, which has a certain similarity with the local pattern of *horizontal pattern*, so the model is easy to confuse these two materials. In addition, due to the similar roughness between *rough towel* and *sponge*, 31% of the M4 (sponge) samples are still misclassified as M1 (rough towel) in the material recognition of 10 N sampled data by DenseNet. MRFE has satisfactory robustness and can still effectively extract category features even when the change of sampling intensity causes the change of tactile features, thus maintaining the overall accuracy of up to 95%. In this experiment, the overall performance of MRFE is better than that of DenseNet.

The cross-batch experiments

The cross-batch experiments are conducted on three different batches of tactile data. Each group of experiments is obtained by training on one batch of data and testing on the remaining

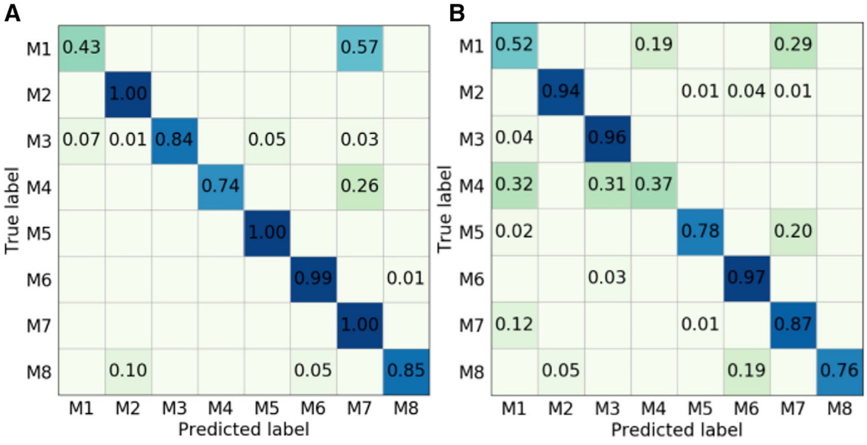


Figure 9. The confusion matrices obtained by MRFE trained on *Batch* 2 data

(A) The confusion matrix on the *Batch* 1 data.

(B) The confusion matrix on the *Batch* 3 data.

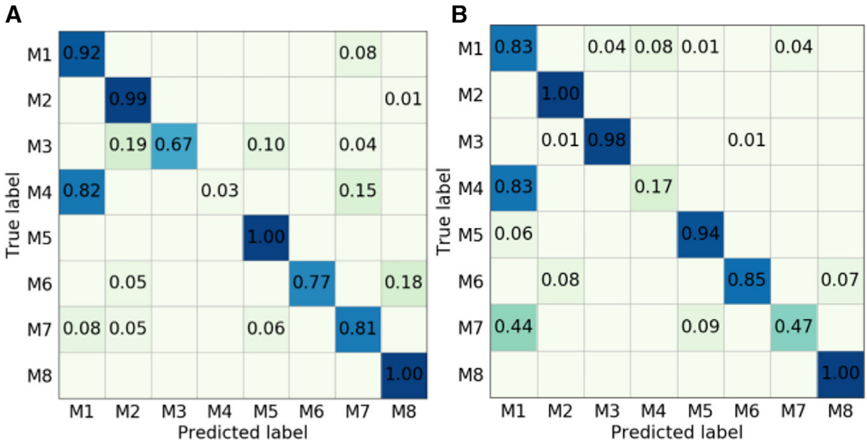


Figure 10. The confusion matrices obtained by MRFE trained on *Batch* 3 data

(A) The confusion matrix on the *Batch* 1 data.

(B) The confusion matrix on the *Batch* 2 data.

two batches. Furthermore, we split each training data into the training and validation sets at 2:1 proportion.

From Table 5, it is found that the models trained on different batches perform differentially. For example, the accuracy of ShuffleNet in *Batch* 1 validation set is 99.30%, whereas it drops sharply to only 46.43% on *Batch* 3 data. This may be caused by systematic errors in the disassembly and assembly process of the robotic arm, which leads to a considerable difference in different batches. DenseNet's performance is superior to other models, but its accuracy on the *Batch* 3 data only reaches 66.83%. In comparisons, GCPL and STAM perform better than those of DenseNet on *Batch* 2 and *Batch* 3. Further, MRFE gets the best performance with 88.50% and 75.50% accuracy scores, respectively, on the *Batch* 2 and *Batch* 3 data. This means that MRFE can effectively correct the class overlapping problem in the prediction and improve the material recognition accuracy in other different batches. The results reveal that the difference among different batches seriously deteriorate the performance of all models, making the classification task even harder. Therefore, different conditions on data collection process remain great challenges for the tactile recognition task. Whereas MRFE performs best in all batches.

From Figures 8, 9, and 10, we can find the more specific recognition accuracy of each material category. In Figure 8, the proposed MRFE model trained with *Batch* 1 data obtains higher accuracy rates in almost all categories on the other two batches. The worst result is M7 (*gauze*), which gets an accuracy score of 35% on the *Batch* 3 data. The reason is that *gauze* and *coarse towel* are very similar in tactile roughness, making the classification results poorer than those of other classes.

Table 6. Texture classification accuracy scores

Models	*Batch* 2_val	*Batch* 1	*Batch* 3
ShuffleNet	98.21 ± 1.12	60.38	47.17
ResNet	99.64 ± 0.06	84.07	50.35
DenseNet	99.31 ± 0.01	80.51	75.26
GCPL	99.82 ± 0.01	83.18	76.75
STAM	99.53 ± 0.01	84.75	74.96
MRFE(ours)	99.88 ± 0.02	85.63	77.13

From Table 6, it is found that the performance of these models trained on *Batch* 2 data also declines when transferred to *Batch* 1 or *Batch* 3 data. It is worth noting that the best accuracy of these models trained on *Batch* 2 data gets a higher score of 77.13% on the *Batch* 3 data compared with that of the model trained on *Batch* 1 data in Table 5 (75.50%). This is because *Batch* 2 and *Batch* 3 data have the same fixed sampling pressure, whereas *Batch* 1 data have random pressure, and the exploratory motions of sampling change from press to slide.

As shown in Figure 9, our model performs well with higher accuracy than 75% scores in most cases, except for the M1 class (*coarse towel*) and the M4 class (*sponge*). It is found that some samples in M4 are misclassified to M1 (*coarse towel*), M3 *(relief cloth*), and M7 (*gauze*). Figure 3 shows that the similarity of *sponge* samples of *Batch* 2 and *Batch* 3 is lower than 0.3, which brings great difficulty to the cross-batch prediction. In addition, the similar rough tactile impression of M1 (*coarse towel*), M4 (*sponge*), and M7 (*gauze*) is also the reason for the decline in the overall accuracy.

As shown in Table 7, the performance of ShuffleNet and ResNet models decreases greatly in the prediction of other batches of datasets. In contrast, GCPL and STAM perform better with higher accuracy scores on both *Batch* 2 and *Batch* 3, whereas MRFE maintains relatively high generalization performance, with an accuracy at 77.38% and 78% in *Batch* 1 and *Batch* 2 data, respectively. In Figure 10, the performance of MRFE on M4 (*sponge*) on *Batch* 1 and *Batch* 2 data is 3% and 17%, respectively. As mentioned earlier, the similar tactile roughness of *sponge* and *coarse towel* determines the indiscernibility.

Ablation studies

In MRFE, two modules are used: spatial attention enhancement (SAE) and coordinate attention enhancement (CAE). The effectiveness of SAE and CAE in MRFE is verified based on the ablation experiments.

It should be noted that SAE is used to extract key spatial features, providing the attention weights from spatial dimensions, and CAE acquires the global characteristics in the horizontal and vertical directions. Because of different working principle, SAE and CAE would extract different features and output diverse predictions for decisions. The experiments are conducted on the cross-velocity data and the cross-batch data, respectively.

Table 7. Texture classification accuracy scores

Models	*Batch* 3_val	*Batch* 1	*Batch* 2
ShuffleNet	99.33 ± 0.52	44.57	60.21
ResNet	99.70 ± 0.05	46.17	60.25
DenseNet	99.61 ± 0.00	65.20	71.79
GCPL	99.84 ± 0.01	68.23	75.94
STAM	99.79 ± 0.01	72.86	76.73
MRFE(ours)	99.91 ± 0.03	77.38	78.00

Table 9. Texture classification accuracy scores

Models	*2+4+6 cm/s*_val	*8 cm/s*
MRFE(wo SAE)	97.92 ± 0.02	95.65
MRFE(wo CAE)	99.04 ± 0.01	96.23
MRFE(ours)	100.00 ± 0.00	97.75

The cross-velocity experiment is set as in The cross-batch experiments section. In Table 8, MRFE(wo SAE) and MRFE(wo CAE) represent the case in which our model work without the SAE module and CAE module, respectively.

It is found from Table 8 that when the MRFE models (with or without SAE and CAE) are trained and tested on the dataset gathered at the same velocity, their accuracy scores are all higher than 98%. The reason mainly lies in that the data distribution of those collected at the same velocity is similar, so it is a relatively easy task. And using a single module would affect the performance of MRFE, which is not kept at the same level when testing at different velocities. The diversity of data obtained at different velocities makes it a hard task for each module to handle solely. However, due to the different working principles, the SAE and CAE modules extract diverse features and produce different predictions. Consequently, by combining diverse decisions, MRFE can produce more robust results. Although its performance would still drop when testing on data of different velocities, it still exhibits higher generalization ability and produces the highest prediction results compared with those using a module solely.

A similar conclusion can be found in Table 9 when combining the data of three different velocities as the training data. It is obvious that with more training data, both MRFE(wo SAE) and MRFE(wo CAE) perform much better compared with the previous case. And their performances are higher than that of STAM. However, with a single module, MRFE cannot perform better than GCPL. In contrast, with both modules, MRFE outperforms other models, confirming that the fusion of two diverse modules promises the highest generalization ability.

Table 10 lists the results obtained by the cross-batch setting. It is found that the models trained on different batches perform differentially. As observed before, each model performs well on *Batch* 1 validation set, but the results drop sharply on *Batch* 3 data due to the system drift in the disassembly and assembly process of the robotic arm. The use of a single module still leads to slightly lower results. And the deployment of both modules gets the best performance on both *Batch* 2 and *Batch* 3 data, showing that the construction of MRFE can better handle the material recognition problem with higher generalization ability in the case of cross-batch situation.

In short, the success of our model mainly lies in the fusion of SAE and CAE modules. Owing to the different attention mechanisms, SAE and CAE can catch different key features and produce diverse and accurate predictions. Consequently, the fusion of the two modules promises higher performance and leads to more robust results in hard tasks, such as the cross-batch data prediction.

Conclusions

We discuss the tactile data recognition problem by collecting three batches of data under changing installation environments, including the changes in force, speed, pose, and so on. The SSIM measure shows that three batches of data are of great distances. To handle the cross-batch tactile recognition problem, we propose the MRFE, which combines the spatial attention enhancement module and coordinates attention enhancement module to extract effective features. The experimental results show that the proposed method can correct the feature offset effectively, reaching 88.50% in the best accuracy of cross-batch generalization and 75.50% in the worst case.

The shortcomings of our approach mainly lie in two aspects: (1) the eight types of test materials are all fabric-based materials, so the data collected in this study are not applicable to other materials, such as wood and metal. To enhance the research work to real-world applications, more types of materials should be tried in the future. Furthermore, there are more experimental settings that should be extensively studied, including the use of different types of e-skin and the surface of the data collection unit. (2) The cross-batch data analysis can be well extended by considering the difference of feature spaces among different batches, so the application of transfer learning methods would be a promising solution.

Therefore, in the future, we will try to apply the transfer learning technique to discover the invariance features across different batches and deal with more diverse generalization challenges in the field of robotics.

Table 8. Texture classification accuracy scores

Models	*2 cm/s*_val	*4 cm/s*	*6 cm/s*	*8 cm/s*
MRFE(wo SAE)	98.31 ± 0.02	83.46	78.74	78.15
MRFE(wo CAE)	99.65 ± 0.01	84.52	80.79	78.46
MRFE(ours)	100.0 ± 0.00	90.25	84.63	83.25

Table 10. Texture classification accuracy scores

Models	*Batch* 1_val	*Batch* 2	*Batch* 3
MRFE(wo SAE)	99.25 ± 0.02	85.41	73.81
MRFE(wo CAE)	99.71 ± 0.01	84.33	74.07
MRFE(ours)	100.00 ± 0.00	88.50	75.50

Limitations of the study

The limitation of this study mainly lies in the types of materials used for tactile recognition, which are solely fabrics. Since materials possess diverse surface characteristics, experiments on fabrics alone cannot well simulate the open scenes. Furthermore, the data collection process only involves planar surface, and the real-world application may also need to consider the non-planar case. Besides, surface roughness is also important for perception. In short, more efforts should be made to collect data under various experiment settings in the future.

RESOURCE AVAILABILITY

Lead contact

Further information and requests should be directed to the lead contact, Dr. Kunhong Liu (lkhqz@xmu.edu.cn).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- The manuscript provided the dataset at https://github.com/MLDMXM2017/MRFE-NN.
- The related codes and models are available at https://github.com/MLDMXM2017/MRFE-NN.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

ACKNOWLEDGMENTS

This work has been supported by National Natural Science Foundation of China (No. 52375027), the public technology service platform project of Xiamen City (No.3502Z20231043), and Open Fund of Hubei Key Laboratory of Mechanical Transmission and Manufacturing Engineering at Wuhan University of Science and Technology (MTMEOF2019A01). This study does not involve any ethical experiments and does not require any approval by the Ethics Review Committee.

AUTHOR CONTRIBUTIONS

H.F., Q.Y., and X.H. designed the algorithm, conducted computational experiments, and wrote the manuscript. K.L. and Y.X. revised the manuscript and supervised the project. All authors approved the final manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- METHOD DETAILS
  - ○ Methodology
  - ○ Overall structure
  - ○ Multi-receptive field attention enhancement
  - ○ Uncertainty weighting
- QUANTIFICATION AND STATISTICAL ANALYSIS
- ADDITIONAL RESOURCES

Received: December 4, 2024
Revised: January 15, 2025
Accepted: March 28, 2025
Published: April 2, 2025

REFERENCES

1. Liu, X., Yang, W., Meng, F., and Sun, T. (2024). Material Recognition Using Robotic Hand With Capacitive Tactile Sensor Array and Machine Learning. IEEE Trans. Instrum. Meas. *73*, 1–9. https://doi.org/10.1109/TIM.2024.3383886.
2. Zhao, H., Zhang, Y., Han, L., Qian, W., Wang, J., Wu, H., Li, J., Dai, Y., Zhang, Z., Bowen, C.R., and Yang, Y. (2023). Intelligent Recognition Using Ultralight Multifunctional Nano-Layered Carbon Aerogel Sensors with Human-Like Tactile Perception. Nano-Micro Lett. *16*, 11. https://doi.org/10.1007/s40820-023-01216-0.
3. Ma, F., Li, Y., Chen, M., and Yu, W. (2023). A data-driven robotic tactile material recognition system based on electrode array bionic finger sensors. Sensor Actuator Phys. *363*, 114727. https://doi.org/10.1016/j.sna.2023.114727.
4. Rasouli, M., Chen, Y., Basu, A., Kukreja, S.L., and Thakor, N.V. (2018). An extreme learning machine-based neuromorphic tactile sensing system for texture recognition. IEEE Trans. Biomed. Circuits Syst. *12*, 313–325.
5. de Farias, C., Marturi, N., Stolkin, R., and Bekiroglu, Y. (2021). Simultaneous Tactile Exploration and Grasp Refinement for Unknown Objects. IEEE Robot. Autom. Lett. *6*, 3349–3356.
6. Liu, Y., Wang, J., Liu, T., Wei, Z., Luo, B., Chi, M., Zhang, S., Cai, C., Gao, C., Zhao, T., et al. (2025). Triboelectric tactile sensor for pressure and temperature sensing in high-temperature applications. Nat. Commun. *16*, 383. https://doi.org/10.1038/s41467-024-55771-0.
7. Nottensteiner, K., Sachtler, A., and Albu-Schffer, A. (2021). Towards Autonomous Robotic Assembly: Using Combined Visual and Tactile Sensing for Adaptive Task Execution. J. Intell. Rob. Syst. *101*, 1–22.
8. Chathuranga, D.S., and Hirai, S. (2013). Investigation of a Biomimetic Fingertip's Ability to Discriminate Fabrics Based on Surface Textures (IEEE), pp. 1667–1674.
9. Baishya, S.S., and Bäuml, B. (2016). Robust Material Classification with a Tactile Skin Using Deep Learning (IEEE), pp. 8–15.
10. Taunyazov, T., Chua, Y., Gao, R., Soh, H., and Wu, Y. (2020). Fast Texture Classification Using Tactile Neural Coding and Spiking Neural Network (IEEE), pp. 9890–9895.
11. Huang, S., and Wu, H. (2021). Texture Recognition Based on Perception Data from a Bionic Tactile Sensor. Sensors *21*, 5224.
12. Sankar, S., Brown, A., Balamurugan, D., Nguyen, H., Iskarous, M., Simcox, T., Kumar, D., Nakagawa, A., and Thakor, N. (2019). Texture Discrimination Using a Flexible Tactile Sensor Array on a Soft Biomimetic Finger (IEEE), pp. 1–4.
13. Kaboli, M., and Cheng, G. (2018). Robust tactile descriptors for discriminating objects from textural properties via artificial robotic skin. IEEE Trans. Robot. *34*, 985–1003.
14. Wang, R., Hu, S., Zhu, W., Huang, Y., Wang, W., Li, Y., Yang, Y., Yu, J., and Deng, Y. (2023). Recent progress in high-resolution tactile sensor array: From sensor fabrication to advanced applications. Prog. Nat. Sci.: Mater. Int. *33*, 55–66. https://doi.org/10.1016/j.pnsc.2023.02.005.
15. Chen, C., Liu, H., Zhu, X., Wu, D., and Xie, Y. (2019). The impact of the electronic skin substrate on the robotic tactile sensing. Int. J. Human. Robot. *16*, 1950026.
16. Liu, K., Yang, Q., Xie, Y., and Huang, X. (2023). Towards Open-Set Material Recognition Using Robot Tactile Sensing (IEEE), pp. 10345–10351.
17. Woodburn, J., and Helliwell, P.S. (1996). Observations on the F-Scan in-shoe pressure measuring system. Clin. Biomech. *11*, 301–304.
18. Price, C., Parker, D., and Nester, C. (2016). Validity and repeatability of three in-shoe pressure measurement systems. Gait Posture *46*, 69–74.

19. Guo, W.-T., Lei, Y., Zhao, X.-H., Li, R., Lai, Q.-T., Liu, S.-Z., Chen, H., Fan, J.-C., Xu, Y., Tang, X.-G., et al. (2024). Printed-scalable microstructure BaTiO3/ecoflex nanocomposite for high-performance triboelectric nanogenerators and self-powered human-machine interaction. Nano Energy *131*, 110324. https://doi.org/10.1016/j.nanoen.2024.110324.

20. Zhao, X.-H., Lai, Q.-T., Guo, W.-T., Liang, Z.-H., Tang, Z., Tang, X.-G., Roy, V.A.L., and Sun, Q.-J. (2023). Skin-Inspired Highly Sensitive Tactile Sensors with Ultrahigh Resolution over a Broad Sensing Range. ACS Appl. Mater. Interfaces *15*, 30486–30494. https://doi.org/10.1021/acsami.3c04526.

21. Zhang, S., Yang, Y., Shan, J., Sun, F., Xue, H., and Fang, B. (2024). PaLmTac: A Vision-Based Tactile Sensor Leveraging Distributed-Modality Design and Modal-Matching Recognition for Soft Hand Perception. IEEE J. Sel. Top. Signal Process. *18*, 288–298. https://doi.org/10.1109/JSTSP.2024.3386070.

22. Fishel, J.A., and Loeb, G.E. (2012). Bayesian exploration for intelligent identification of textures. Front. Neurorobot. *6*, 4.

23. Xu, D., Loeb, G.E., and Fishel, J.A. (2013). Tactile Identification of Objects Using Bayesian Exploration (IEEE), pp. 3056–3061.

24. Fang, B., Long, X., Sun, F., Liu, H., Zhang, S., and Fang, C. (2022). Tactile-Based Fabric Defect Detection Using Convolutional Neural Network With Attention Mechanism. IEEE Trans. Instrum. Meas. *71*, 1–9. https://doi.org/10.1109/TIM.2022.3165254.

25. Fishel, J.A., and Loeb, G.E. (2012). Sensing Tactile Microvibrations with the BioTac—Comparison with Human Sensitivity (IEEE), pp. 1122–1127.

26. Metta, G., Sandini, G., Vernon, D., Natale, L., and Nori, F. (2008). The iCub Humanoid Robot: An Open Platform for Research in Embodied Cognition (ACM), pp. 50–56.

27. Taunyazov, T., Koh, H.F., Wu, Y., Cai, C., and Soh, H. (2019). Towards Effective Tactile Identification of Textures Using a Hybrid Touch Approach (IEEE), pp. 4269–4275.

28. Chen, L., Zhu, Y., and Li, M. (2024). Tactile-GAT: tactile graph attention networks for robot tactile perception classification. Sci. Rep. *14*, 27543. https://doi.org/10.1038/s41598-024-78764-x.

29. Yang, Q., Liu, K., Xie, Y., and Huang, X. (2023). Drop to Transfer: Learning Transferable Features for Robot Tactile Material Recognition in Open Scene. IEEE Trans. Instrum. Meas. *72*, 1–11. https://doi.org/10.1109/TIM.2023.3243664.

30. I-Scan System. http://www.tekscan.com/products-solutions/systems/i-scan-system.

31. Xie, Y., Chen, C., Wu, D., Xi, W., and Liu, H. (2019). Human-Touch-Inspired Material Recognition for Robotic Tactile Sensing. Appl. Sci. *9*, 2537.

32. Wang, Z., Bovik, A.C., Sheikh, H.R., and Simoncelli, E.P. (2004). Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. *13*, 600–612.

33. Dunn, J.C. (1974). Well-separated clusters and optimal fuzzy partitions. J. Cybern. *4*, 95–104.

34. Zhang, X., Zhou, X., Lin, M., and Sun, J. (2018). Shufflenet: An Extremely Efficient Convolutional Neural Network for Mobile Devices (IEEE), pp. 6848–6856.

35. He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition (IEEE), pp. 770–778.

36. Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K.Q. (2017). Densely Connected Convolutional Networks (MDPI), pp. 4700–4708.

37. Yang, H.M., Zhang, X.Y., Yin, F., and Liu, C.L. (2018). Robust Classification with Convolutional Prototype Learning (IEEE), pp. 3474–3482.

38. Cao, G., Zhou, Y., Bollegala, D., and Luo, S. (2020). Spatio-temporal Attention Model for Tactile Texture Recognition (Taylor & Francis), pp. 9896–9902.

39. Holschneider, M., Kronland-Martinet, R., Morlet, J., and Tchamitchian, P. (1990). A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform (Springer), pp. 286–297.

40. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A.L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans. Pattern Anal. Mach. Intell. *40*, 834–848.

41. Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., and Cottrell, G. (2018). Understanding Convolution for Semantic Segmentation (IEEE), pp. 1451–1460.

42. Mehta, S., Rastegari, M., Caspi, A., Shapiro, L., and Hajishirzi, H. (2018). Espnet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation (IEEE), pp. 552–568.

43. Hou, Q., Zhou, D., and Feng, J. (2021). Coordinate Attention for Efficient Mobile Network Design (IEEE), pp. 13713–13722.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and algorithms		
Scikit-learn	Version 0.24.2	https://pypi.org/project/scikit-learn/0.24.2/
Python	Version 3.9.17	https://www.python.org/downloads/release/python-3917/
Matplotlib	Version 3.6.3	https://matplotlib.org/3.6.3/
Data	The data of this study.	https://github.com/MLDMXM2017/MRFE-NN
Code	The code for training and test of our model.	https://github.com/MLDMXM2017/MRFE-NN
Other Items	The other items.	https://github.com/MLDMXM2017/MRFE-NN

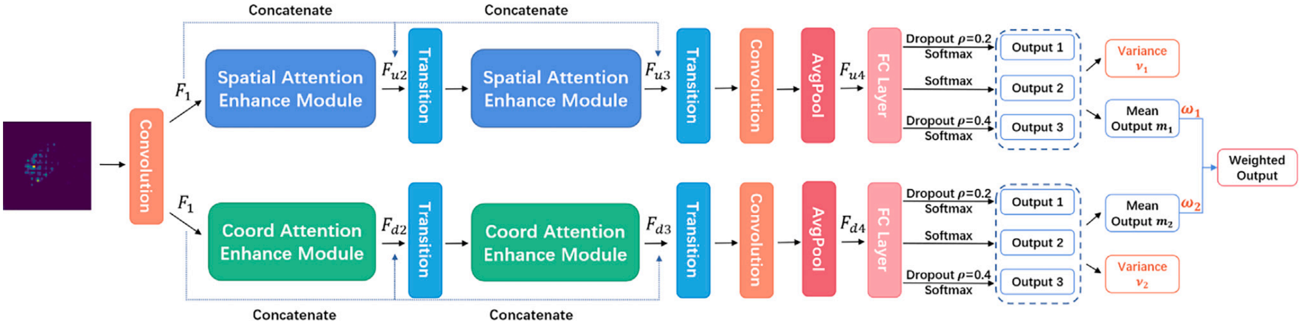
METHOD DETAILS

Methodology

To deal with feature offsets in open scenes, this paper proposes multi-receptive field attention enhancement network (MRFE) to reinforce important feature subsets.

Overall structure

As shown in Figure 11, the input image first passes through a 3 × 3 convolution layer to obtain the shared feature F_1. Then the network is divided into two branches, and the input of each branch is F_1. The attention feature subsets F_{u2} and F_{u3} are extracted by two consecutive spatial attention enhancement modules and transition layers. F_{u2}, and F_{u3} are concatenated into subsequent convolution layers and average pooling layer to obtain the flattened feature vector F_{u4}. To enhance the generalization of the proposed model and quantify the uncertainty of prediction, dropout is adopted on F_{u4}. And the neuron's deletion ratio ρ is set to 0, 0.2, and 0.4 respectively to increase the perturbation and then input into the fully connected layer with Softmax to obtain more diverse prediction results. The mean output m_1 is calculated by the three prediction outputs. The variance is calculated as the uncertainty v_1 of this group of predictions. Similarly, attention feature subsets F_{d2} and F_{d3} are extracted by two consecutive coordinate attention enhancement modules and transition layers, and then combined with shared feature F_1 to send into convolution layers and average pooling layer to obtain flattened feature vector F_{d4}. Also, the mean output m_2 and variance v_2 of the three outputs are calculated by using different dropout ratios for F_{d4}.



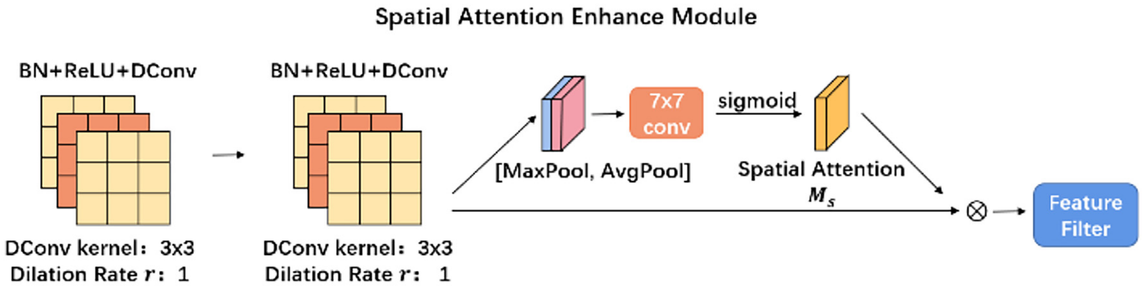
Overall structure of MRFE

Multi-receptive field attention enhancement

The dilated convolution,[39] originally derived from the porous wavelet transform of digital signal processing, was later widely used in the semantic segmentation problem,[40–42] which can maintain the desired feature map resolution while expanding the receptive field, thus replacing down-sampling and up-sampling operations. The dilated rate r represents the stride length when sampling the input sample, and the standard convolution is a special case of $r = 1$. To adapt to the material recognition task with different grain sizes of texture features, 3 × 3 dilated convolution layers with dilated rate $r = 1$ and $r = 2$ are used in the space and coordinate attention enhancement modules respectively, to change the sampling distance from the center point and obtain multi-scale feature maps under different receptive fields.

To extract key feature subsets, we deployed a spatial attention enhancement module to calculate the attention weights from spatial dimensions, and reweighted the output to achieve adaptive feature refinement. The specific structure is shown in Figure 12. Batch Normalization (BN), ReLU, and dilated convolution are deployed on the input to obtain feature vector $\boldsymbol{F}$. And the average pooling and maximum pooling are also applied respectively to $\boldsymbol{F}$ to extract the global and characteristic information. These two features are concatenated and fed into the 7 × 7 convolution layer. After deploying the sigmoid activation function, the spatial attention figure M_s is obtained, and then calculate tensor product (i.e., $\bigotimes$) between feature $\boldsymbol{F}$ and M_s:

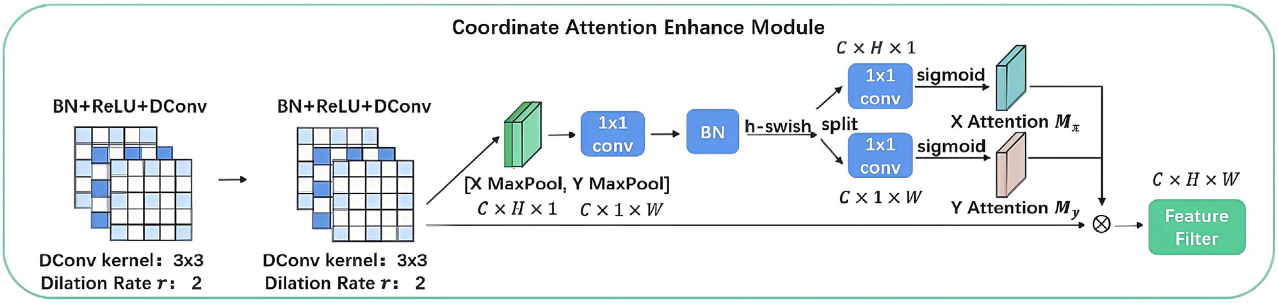
$$F_u = F \bigotimes M_s(F) \quad \text{(Equation 4)}$$



Spatial attention enhancement module

In the tactile material data, some samples of different classes have similar shape features caused by the robotic arm pressing materials. If the deep network focuses on the global shape features, it tends to make misclassification. The fine-grained texture features should be better extracted to facilitate classification. Therefore, we further deploy the coordinate attention enhancement module, with the specific structure shown in Figure 13. The original coordinate attention mechanism[43] adopted the average pooling on the X-axis and the Y-axis to acquire the global characteristics in the horizontal and vertical directions. For the fine-grained texture classification in our study, maximum pooling is more conducive to extract significant texture features and reduces the influence of useless information such as noise. Therefore, in the coordinate attention enhancement module, the feature vector $\boldsymbol{F}$ extracted by dilated convolution is first pooled along the X-axis and Y axis respectively, to obtain texture feature responses in the horizontal and vertical directions as much as possible, then concatenate and input them into the subsequent convolution layer, Batch Normalization (BN) and h-swish activation functions. The acquired features are split to obtain the vector of $C \times H \times 1$ and $C \times 1 \times W$. Finally, through 1 × 1 convolution and sigmoid function to obtain attention maps M_x and M_y on the X and Y axes and the tensor product is calculated with the feature $\boldsymbol{F}$:

$$F_d = F \bigotimes M_x(F) \bigotimes M_y(F) \quad \text{(Equation 5)}$$



Coordinate attention enhancement module

Furthermore, we filter the feature vectors F_u and F_d, according to the attention-reweighted feature value in ascending order, take the elements q_u and q_d in the top 25% as the threshold, compare each element with q_u and q_d, to construct mask vector m_u and m_d with the same dimensions of F_u and F_d. The value of i^{th} elements in m_u and m_d are as follows:

$$m_u(i) = \begin{cases} 0, if\ F_u \leq q_u \\ 1, otherwise \end{cases} \quad \text{(Equation 6)}$$

$$m_d(i) = \begin{cases} 0, if\ F_d \leq q_d \\ 1, otherwise \end{cases} \quad \text{(Equation 7)}$$

Then, retain feature values enhanced by attention mechanisms and eliminate feature values with low attention weight in F_u and F_d by the formula:

$$F_u{}' = F_u \odot m_u \quad \text{(Equation 8)}$$

$$F_d{}' = F_d \odot m_d \quad \text{(Equation 9)}$$

Uncertainty weighting

Generally, the prediction output of a multi-branch neural network can adopt late fusion, that is fusion on the prediction score level. The common late fusion method has the average, maximum, and weighted average, where the weighted average is a more reasonable way, but the weight coefficient often needs to be set manually. Considering that the prediction results given by deep neural networks are not always reliable, we hope that the model can give the prediction results along with corresponding confidence, and carry out weighted fusion according to the prediction confidence to obtain the final output. Therefore, uncertainty needs to be modeled. When different perturbations are added to a model, if the model maintains stable predictions, that is, the variance of each prediction is relatively small, and the model is considered to be less uncertain. On the contrary, if the model is sensitive to perturbations and the prediction results of the same input change greatly, the model is likely to make wrong predictions with great uncertainty.

Based on this, different proportions of dropouts are used for output feature vectors F_{u4} and F_{d4} from two paths. By calculating mean outputs m_1, m_2 and variances v_1, v_2, to obtain the reweighted predictions. The weight coefficient is calculated as follows:

$$\omega_1 = \frac{e^{v_2}}{e^{v_1}+e^{v_2}} \quad \text{(Equation 10)}$$

$$\omega_2 = \frac{e^{v_1}}{e^{v_1}+e^{v_2}} \quad \text{(Equation 11)}$$

As can be seen from Equations 7 and 8, the output from the path with greater uncertainty will obtain a smaller weight coefficient, while the output from another path with less uncertainty will obtain a larger weight. The final prediction result of the model is $\omega_1 m_1 + \omega_2 m_2$.

QUANTIFICATION AND STATISTICAL ANALYSIS

We use Accuracy score to evaluate the overall prediction performance of the model, which indicates the model's ability to correctly identify samples while minimizing errors. The metric is defined by:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{(Equation 12)}$$

ADDITIONAL RESOURCES

This study generates three batches of tactile data, which is available online at: https://github.com/MLDMXM2017/MRFE-NN.