

LAPORAN TUGAS AKHIR

MK PEMROSESAN SINYAL MULTIMEDIA

DIKUMPULKAN PADA: 20 JUNI 2023

Nama Mahasiswa : Muhammad Aidan Daffa Junaidi, Fatma Putri Ramadhani
NPM : 1906300800, 1906381735
Judul Paper : CNN-based Methods for Object Recognition with High-Resolution Tactile Sensors
Penulis : Juan M. Gandarias, Alfonso J. García-Cerezo, Jesús M. Gómez-de-Gabriel
Dipublikasikan di : IEEE ; arXiv
Tahun : 2019 ; 2023
Modality/Alat Pencitraan : Komputer
Task/Tujuan pengolahan : Membandingkan Algoritma Machine Learning dalam Permasalahan Object Recognition pada High-Resolution Tactile Sensors
Bahasa Pemrograman : Matlab
Link Paper : <https://paperswithcode.com/paper/cnn-based-methods-for-object-recognition-with>
Link Source Code : https://github.com/TalSLab/CNN-based-Methods-for-Tactile-Object-Recognition/blob/master/Squeeze_NN.m

RESUME PAPER

1 PENDAHULUAN

Jelaskan dalam Bahasa Indonesia:

- a) latar belakang masalah/topik penelitian yang diangkat,

Kemampuan peraba dibutuhkan oleh manusia salah satunya untuk melakukan object recognition. Informasi taktil atau peraba juga dibutuhkan oleh sistem robotik. Maka dari itu, terdapat 2 masalah terkait persepsi taktil dalam bidang robotik, yaitu mendapatkan informasi indra peraba dan kemampuan untuk memproses informasi tersebut. Untuk mengenali objek yang bersentuhan melalui tactile sensors, Convolutional Neural Network (CNN) dapat digunakan. Selain itu, CNN juga dapat digunakan untuk melakukan transfer knowledge dari satu domain dan melakukan training pada network baru (transfer learning).

Tetapi, pengaplikasian CNN biasa kepada tactile object recognition tidak dapat dilakukan efektif karena resolusi dan feature lain pada foto tactile object lebih rendah dibandingkan RGB image. Sehingga, paper ini berfokus pada pengaplikasian persepsi tactile untuk robot yang memiliki kemampuan komputasi dan energi yang terbatas. Paper ini menjelaskan perbandingan hasil CNN lain untuk mengenali pressure image. Terdapat 2 pendekatan, yaitu transfer learning dengan CNN pre-trained dataset dari RGB image dan training custom CNN dari awal dengan informasi tactile saja. Custom CNN ini diberi nama TactNet dengan 3 konfigurasi, yaitu TactNet-4, TactNet-6, dan TactResNet.

- b) urgensi topik penelitian,

Topik penelitian ini dibutuhkan untuk mendapatkan metode object recognition foto tactile lebih baik dibandingkan CNN biasa dan untuk membandingkan metode mana yang paling menguntungkan dalam mengenali foto tactile object.

- c) metode-metode terkini (state-of-the-art) yang telah dihasilkan oleh peneliti lain yang disebutkan dalam paper tsb,

Nomor referensi	Penulis	Judul Paper	Metode yang diusulkan	Kelebihan yang diusung	Celah/Kekurangan
41	Alexander Schmitz; Yusuke Bansho; Kuniaki Noda; Hiroyasu Iwata; Tetsuya	Tactile object recognition using deep learning and dropout	Penggunaan deep learning dengan dropout	Recognition rate meningkat menggunakan multimodal	Menggunakan PCA tidak meningkatkan recognition rate

	Ogata; Shigeki Sugano			sensor information	
43	Pietro Falco; Shuang Lu; Andrea Cirillo; Ciro Natale; Salvatore Pirozzi; Dongheui Lee	Cross-modal visuo-tactile object recognition using robotic active exploration	Cross-modal visuo-tactile	Lebih baik dengan menggabungkan 2 descriptor, seperti SHOT dan ESF	Menggunakan descriptor dari komunitas computer vision hanya mendapatkan akurasi 50%

d) ide utama dari metode yang penulis usulkan,

Penulis mengusulkan TactNet, yaitu pengklasifikasi tactile information menggunakan CNN. Konfigurasi menggunakan 4 dan 6 plain layer dari arsitektur AlexNet dan 6 layer termasuk residual convolution dari struktur ResNet. TactNet dibuat dan ditraining dari pressure image dataset sehingga membutuhkan tactile data yang sangat banyak.

e) tujuan penelitian/kontribusi/kebaruan yang ditawarkan.

Kebaruan yang ditawarkan dalam paper ini adalah object recognition tactile image menggunakan transfer learning dari CNN sehingga akan lebih efektif dibandingkan dengan RGB image

2 METODE YANG DITAWARKAN

11.1) Prinsip dasar pengolahan sinyal/citra (yang dipelajari pada kuliah ini) yang diterapkan pada metode tsb\

TactNet akan mengumpulkan informasi dari learning process. Untuk mengumpulkan informasi ini, tactile sensor ditempatkan di tempat yang berbeda pada objek secara manual. Kemudian data ini dijadikan sebagai training data. TactNet menggunakan 4 dan 6 layer yang akan dijelaskan di bagian eksperimen.

11.2) Algoritma dan/atau alur kerja metode tsb

Network dari TactNet terdiri dari sejumlah convolutional atau residual convolutions yang belajar untuk mengekstrak tactile feature dari pressure image. Setelah setiap layer convolution, terdapat normalization layer dengan $\epsilon = 10^{-4}$ diikuti dengan rectified linear unit (ReLU) untuk non-linearities. Beberapa layer juga diselengi dengan max-pooling layer dengan stride 2. Terakhir terdapat fully connected layers yang mempelajari untuk mengklasifikasikan input data.

TactNet model ditrain menggunakan teknik data augmentasi dengan melakukan translasi, rotasi, dan berbagai scale sehingga model dapat mengenali objek tanpa bergantung lokasi, orientasi, dan contact pressure-nya.

Kemudian performa network dievaluasi untuk menghasilkan nilai akurasi dan dibuat confusion matrixnya untuk mengetahui class mana yang salah prediksinya.

EKSPERIMEN DAN ANALISIS

1 DATA

Data yang digunakan adalah sekumpulan dari tactile images dengan jumlah 1100 foto yang dibagi menjadi 22 classess dengan label adhesive, allen key, arm, ball, bottle, box, branch, cable, cable pipe, caliper, can, finger, hand, highlighter pen, key, pen, pliers, rock, rubber, scissors, sticky tape and tube. Training set berisi 704 foto, validation test berisi 176 foto, dan testing set berisi 220 foto.

2 KODE

Perbedaan file TactNet4 TactNet6, fourth_Res, dan half_Res :

- TactNet4 terdapat 3 layer konvolusi dan 1 layer fc.
- TactNet6 terdapat 5 layer konvolusi dan 1 layer fc.

- fourth_Res dan half_Res memiliki perbedaan dari segi input data yang digunakan yang setengah dan seperempat lebih kecil resolusinya.

Penjelasan Kode :

```
%% TactNet-4 Network Architecture
TactNet = [
    %input
    imageInputLayer([28 50 1], 'Name', 'Input')

    %conv1
    convolution2dLayer([5 5],8,'Padding','same','Name','conv1')
    batchNormalizationLayer('Name','batch1')
    reluLayer('Name','ReLU1')
    maxPooling2dLayer(2,'Stride',2,'Padding','same','Name','pool1')

    %conv2
    convolution2dLayer([3 3],16,'Padding','same','Name','conv2')
    batchNormalizationLayer('Name','batch2')
    reluLayer('Name','ReLU2')
    maxPooling2dLayer(2,'Stride',2,'Padding','same','Name','pool2')

    %conv3
    convolution2dLayer([3 3],32,'Padding','same','Name','conv3')
    batchNormalizationLayer('Name','batch3')
    reluLayer('Name','ReLU3')
    maxPooling2dLayer(2,'Stride',2,'Padding','same','Name','pool3')

    %fcl
    fullyConnectedLayer(22,'Name','fcl')
    softmaxLayer('Name','softmax')
    classificationLayer('Name','output')
];
```

Potongan kode diatas merupakan potongan kode yang berfungsi untuk membangun model arsitekturnya. dimulai dari input layer dan diakhiri dengan output layer.

```
%% Download Tactile Data (training and test)
images_dir = fullfile('Experiment_IIEE/Images');

images = imageDatastore(images_dir,'IncludeSubfolders',true,'LabelSource',...
    'foldernames','FileExtensions','.jpg');

tbl = countEachLabel(images);

% Notice that each set now has exactly the same number of images.
countEachLabel(images);

% Split the data in training, valiation and test sets
[trainingSet, testSet] = splitEachLabel(images, 0.8, 'randomize');
[trainingSet, validationSet] = splitEachLabel(trainingSet, 0.8, 'randomize');
```

Potongan kode diatas merupakan potongan kode yang berfungsi untuk menge-load data image. data - data image tersebut disimpan di datastore dengan variabel image. Kemudian diakhiri dengan membagi dataset menjadi data untuk di training dan testing dengan perbandingan 8 : 2. Kemudian data training dibagi menjadi data training dan data validasi dengan perbandingan 8 : 2.

```

%% Data Augmentation
dataAugmenter = imageDataAugmenter( ...
    'RandXReflection',1,...
    'RandYReflection',1,...
    'RandXTranslation',[-3 3], ...
    'RandYTranslation',[-3 3],...
    'RandRotation',[-10,10]);

augmented_trainingSet = augmentedImageSource([28 50], trainingSet,...
    'DataAugmentation', dataAugmenter);
augmented_validationSet = augmentedImageSource([28 50], validationSet,...
    'DataAugmentation', dataAugmenter);
augmented_testSet = augmentedImageSource([28 50], testSet,...
    'DataAugmentation', dataAugmenter);

```

Potongan kode diatas berfungsi untuk data augmentasi. Dengan menggunakan function imageDataAugmenter(), maka ter-generate data - data image sesuai dengan parameter yang diberikan.

```

%% Training options
options = trainingOptions('sgdm', ...
    'MaxEpochs',5000, ...
    'Momentum',0.8,...
    'InitialLearnRate', 0.05,...
    'LearnRateSchedule', 'piecewise', ...
    'LearnRateDropFactor', 0.1, ...
    'LearnRateDropPeriod', 200, ...
    'MiniBatchSize', 4096,...
    'ValidationData',validationSet, ...
    'ValidationFrequency',10, ...
    'VerboseFrequency',1, ...
    'ValidationPatience',3,...
    'Verbose',1,...
    'Plots','training-progress');

```

potongan kode diatas berfungsi untuk mengatur parameter - parameter training

```

%% Training the Network
tic;
TactNet_trained = trainNetwork(augmented_trainingSet,TactNet,options);
training_time = toc;

```

potongan kode diatas berfungsi untuk memulai proses training network. bisa dilihat pada function tainNetwork, parameter yang digunakan adalah augmented_trainingSet(variabel yang menyimpan data training), TactNet(arsitektur yang digunakan), dan options(training parameter yang digunakan yang telah di set pada potongan kode sebelumnya.

```

%% Evaluate the performance of the network
tic
predictedLabels = classify(TactNet_trained,testSet);
testLabels = testSet.Labels;
toc

test_accuracy = sum(predictedLabels==testLabels)/numel(predictedLabels)

% Tabulate the results using a confusion matrix.
confMat = confusionmat(testLabels, predictedLabels);

% Convert confusion matrix into percentage form
confMatrix = bsxfun(@rdivide,confMat,sum(confMat,2));

% Plot the confusion matrix
Plot_ConfMatrix(confMatrix);

% Calculate the recognition rate
recognition_rate = mean(diag(confMatrix));

%% Save the workspace
save('training_TactNet4_ws')

```

Potongan kode diatas berfungsi untuk mengevaluasi peforma model yang sudah dibuat pada proses training. dilanjut dengan pembuatan confusion matrix kemudian penyimpanan model.

3 HASIL, INTERPRETASI HASIL, ANALISIS

Hasil Run Code :

TracNet_4

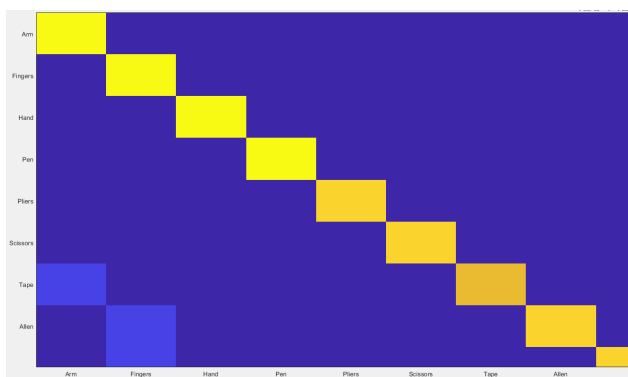
```

>> TactNet_4
Training on single GPU.
Initializing input data normalization.
=====
| Epoch | Iteration | Time Elapsed | Mini-batch | Validation | Mini-batch | Validation | Base Learning |
|       |          | (hh:mm:ss)  | Accuracy   | Accuracy   | Loss       | Loss       | Rate          |
|=====|=====|=====|=====|=====|=====|=====|=====|
| 1     | 1       | 00:00:14    | 5.40%      | 7.95%      | 4.0141     | 3.3767     | 0.0500       |
| 2     | 2       | 00:00:14    | 4.67%      |             | 3.4107     |             | 0.0500       |
|=====|=====|=====|=====|=====|=====|=====|=====|
| 368   | 368    | 00:01:18    | 98.15%     |             | 0.0861     |             | 0.0050       |
| 369   | 369    | 00:01:18    | 96.88%     |             | 0.0942     |             | 0.0050       |
| 370   | 370    | 00:01:18    | 98.86%     | 92.05%     | 0.0815     | 0.2602     | 0.0050       |
|=====|=====|=====|=====|=====|=====|=====|=====|
Elapsed time is 0.652656 seconds.

test_accuracy =

    0.8864

```



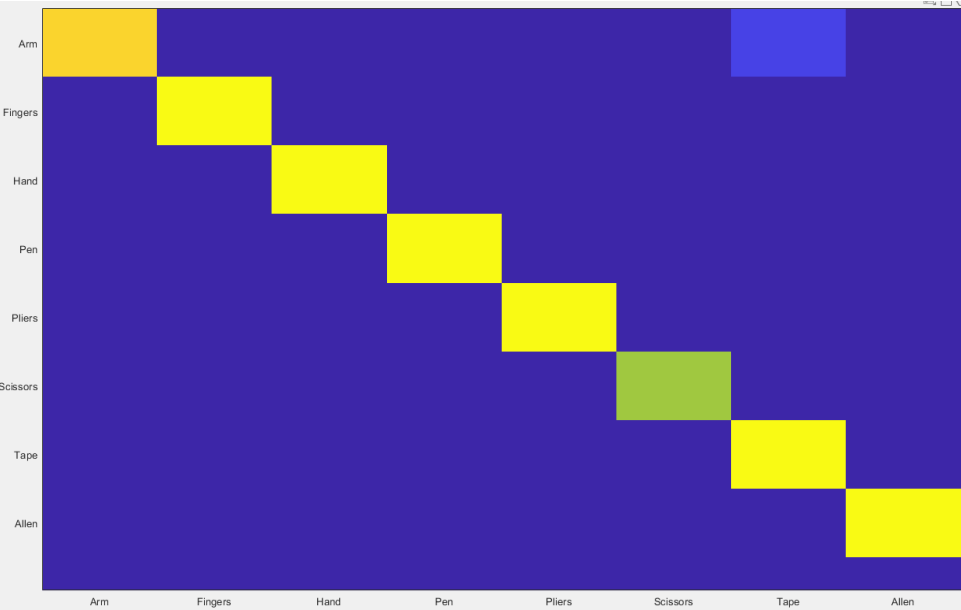
TracNet_6

```
>> TactNet_6
Training on single GPU.
Initializing input data normalization.
=====
| Epoch | Iteration | Time Elapsed | Mini-batch | Validation | Mini-batch | Validation | Base Learning |
|       |          | (hh:mm:ss)  | Accuracy   | Accuracy   | Loss       | Loss       | Rate         |
|=====|=====|=====|=====|=====|=====|=====|=====|
| 1     | 1       | 00:00:03    | 1.28%      | 6.25%      | 4.1590     | 3.4214     | 0.0100      |
| 2     | 2       | 00:00:03    | 5.82%      |             | 3.4505     |             | 0.0100      |
| 3     | 3       | 00:00:04    | 11.93%     |             | 3.1461     |             | 0.0100      |
|=====|=====|=====|=====|=====|=====|=====|=====|

Elapsed time is 0.367086 seconds.

test_accuracy =

0.8682
```



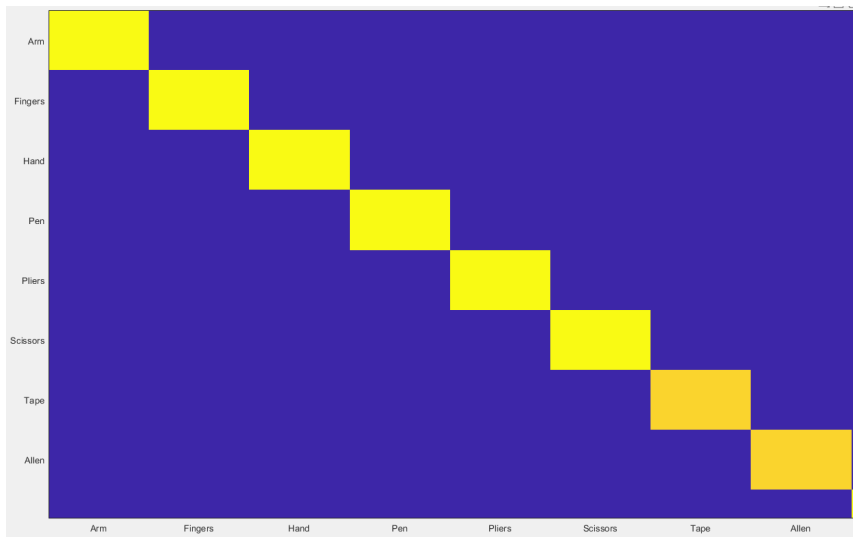
SqueezeNet NN

```
>> Squeeze_NN
Training on single GPU.
Initializing input data normalization.
=====
| Epoch | Iteration | Time Elapsed | Mini-batch | Mini-batch | Base Learning |
|       |          | (hh:mm:ss)  | Accuracy   | Loss       | Rate         |
|=====|=====|=====|=====|=====|=====|
| 1     | 1       | 00:00:01    | 4.69%      | 3.1432     | 0.0010      |
| 5     | 50      | 00:01:30    | 78.12%     | 0.5863     | 0.0010      |
| 10    | 100     | 00:04:15    | 96.88%     | 0.0778     | 0.0010      |
| 14    | 150     | 00:07:04    | 100.00%    | 0.0209     | 0.0010      |
| 15    | 165     | 00:07:53    | 100.00%    | 0.0072     | 0.0010      |
|=====|=====|=====|=====|=====|=====|

Elapsed time is 479.033519 seconds.
Elapsed time is 2.688504 seconds.

test_accuracy =

0.9636
```

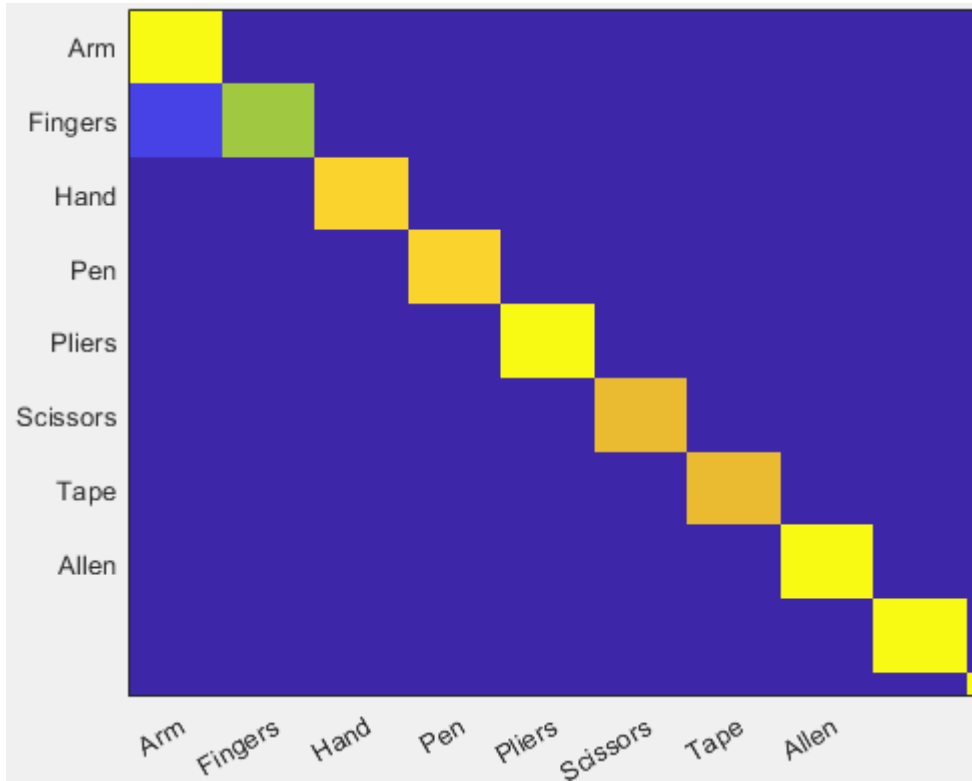


SqueezeNet SVM

```
>> Squeeze_SVM
Elapsed time is 26.983284 seconds.
Elapsed time is 3.375750 seconds.

ans =

    0.8773
```



Berdasarkan hasil run code di atas, akurasi dari setiap metode adalah sebagai berikut

Metode	Akurasi
TactNet-4	88.64%

TactNet-6	86.82%
SqueezeNet NN	96.36%
SqueezeNet SVM	87.73%

TactNet dan SqueezeNet memiliki hasil akurasi yang cukup tinggi, yaitu di atas 86%. Untuk akurasi, nilai tertinggi adalah SqueezeNet NN. Tetapi, untuk waktu komputasi, dapat dilihat bahwa TactNet lebih sebentar dibandingkan SqueezeNet. Hal ini dikarenakan TactNet menggunakan pendekatan lower resolution.

Terdapat beberapa hasil yang didapat pada percobaan berbeda dengan hasil yang dipaparkan di paper, yaitu tingkat akurasi setiap metode. Hal ini dapat terjadi karena tidak adanya parameter yang digunakan oleh penulis paper untuk training optionnya. Maka dari itu, parameter training dapat berbeda yang mengakibatkan hasil yang berbeda juga.

4 ANALISIS TAMBAHAN

Confusion matrix menggunakan colormap parula yang memiliki color scale sebagai berikut



Maka dari itu, jika warna yang didapatkan adalah biru tua, maka tidak ada prediksi TactNet yang memenuhi di foto ekspektasi. Sebaliknya, jika warnanya semakin ke kanan atau semakin kuning, maka terdapat prediksi TactNet yang memenuhi pada foto ekspektasi sehingga pengklasifikasiannya sesuai.

Untuk confusion matrix pada TactNet-4, masih ada beberapa gambar yang salah pengklasifikasiannya. Contohnya adalah arm yang dianggap menjadi tape dan finger yang dianggap menjadi allen key.

Untuk confusion matrix pada TactNet-6 juga masih ada gambar yang salah pengklasifikasiannya, yaitu tape yang dianggap menjadi arm.

Untuk confusion matrix pada SqueezeNet NN, gambar yang diprediksi sesuai dengan ekspektasi yang seharusnya.

Untuk confusion matrix pada SqueezeNet SVM, masih ada gambar yang salah pengklasifikasiannya, yaitu arms yang dianggap menjadi fingers.

KESIMPULAN

Dari hasil percobaan dan analisis paper yang dilakukan, kami mendapatkan kesimpulan bahwa metode TactNet untuk object recognition tactile image memiliki performa yang cukup baik karena waktu komputasinya yang lebih sebentar dibandingkan dengan metode CNN lain. Tetapi, untuk tingkat akurasi metode CNN lain, contohnya SqueezeNet NN, lebih tinggi dibandingkan dengan TactNet. Walaupun begitu, tingkat akurasi TactNet juga sudah baik di atas 86%.

REFERENSI

- [1] Gandarias, J.M., Garcia-Cerezo, A.J. and Gomez-de-Gabriel, J.M. (2019) 'CNN-based methods for object recognition with high-resolution tactile sensors', *IEEE Sensors Journal*, 19(16), pp. 6872–6882. doi:10.1109/jsen.2019.2912968.
- [2] A. Schmitz, Y. Bansho, K. Noda, H. Iwata, T. Ogata and S. Sugano, "Tactile object recognition using deep learning and dropout," 2014 IEEE-RAS International Conference on Humanoid Robots, Madrid, Spain, 2014, pp. 1044-1050, doi: 10.1109/HUMANOIDS.2014.7041493.
- [3] P. Falco, S. Lu, A. Cirillo, C. Natale, S. Pirozzi and D. Lee, "Cross-modal visuo-tactile object recognition using robotic active exploration," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 5273-5280, doi: 10.1109/ICRA.2017.7989619.

[4] A. Khasnobish et al., "Object-shape recognition from tactile images using a feed-forward neural network," The 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, QLD, Australia, 2012, pp. 1-8, doi: 10.1109/IJCNN.2012.6252593.

[5] R. S. Dahiya, G. Metta, M. Valle and G. Sandini, "Tactile Sensing—From Humans to Humanoids," in IEEE Transactions on Robotics, vol. 26, no. 1, pp. 1-20, Feb. 2010, doi: 10.1109/TRO.2009.2033627.

...