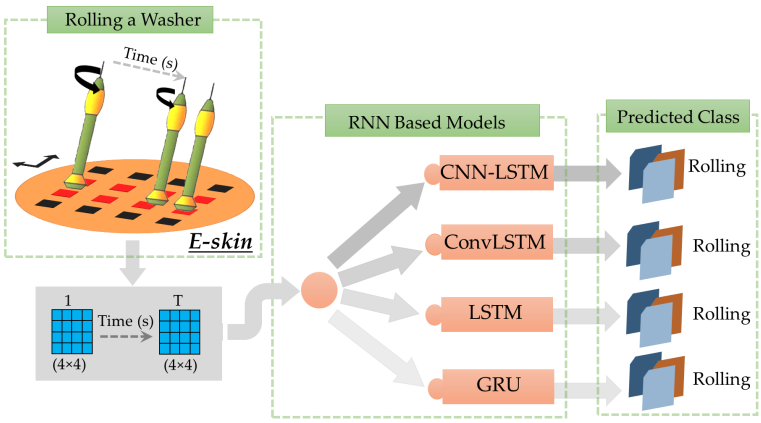
Touch Modality Classification Using Recurrent Neural Networks

Mohamad Alameh, *Student Member, IEEE*, Yahya Abbass, *Student Member, IEEE*, Ali Ibrahim, *Member, IEEE*, Gabriele Moser, *Senior Member, IEEE*, and Maurizio Valle, *Senior Member, IEEE*

***Abstract*— Recurrent Neural Networks (RNNs) are mainly designed to deal with sequence prediction problems and they show their effectiveness in processing data originally represented as time series. This paper investigates the time series characteristics of RNNs to classify touch modalities represented as spatio temporal 3D tensor data. Different approaches are followed in order to propose efficient RNN models aimed at tactile data classification. The main idea is to capture long-term dependence from data that can be used to deal with long sequences represented by employing Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) architectures. Moreover, a case specific approach to dataset organization of the 3D tensor data is presented. The target is to provide efficient hardware-friendly touch modality classification approaches suitable for embedded applications. To this end, the proposed work achieves effective performance in terms of hardware complexity by reducing the FLOPS by 99.98% and the memory storage by 98.34%, with respect to the state-of-art solutions on the same benchmark dataset. This directly affects the time latency and energy consumption of the embedded hardware. Besides, the implemented models shows a classification accuracy higher than those state-of-art solutions. Results demonstrate that the proposed computing architecture is scalable showing acceptable complexity when the system is scaled up in terms of input matrix size and number of classes to be recognized.**



***Index Terms*— Tactile sensing systems, recurrent neural network, deep learning, tactile data classification.**

I. Introduction

Tactile sensing systems attract the research interest in many application domains such as robotics, prosthetic devices, and industrial automation [1], [2]. The main focus is in the areas of sensors and transducers, front end electronics, and smart data processing algorithms. A tactile sensing system is composed of: 1) an array of tactile sensors to sense the applied mechanical stimuli, 2) an interface electronics for signal conditioning and data acquisition, and 3) an embedded digital processing unit for tactile data decoding. The goal is either to mimic the human capabilities in capturing and interpreting tactile data or to respond to the application demands. To be effective, tactile sensors have to sense and extract meaningful information from the contact surface such as force direction and intensity, position, vibrations, objects and texture, or touch modality classification. Such information can be extracted by employing algorithms rooted in machine learning, which have proven their effectiveness in processing tactile data [3].

The adoption of tactile sensing systems in real world application is still limited and challenging [4], [5]. One key aspect is the complexity of the processing algorithms (basically the number of Floating Point Operations - FLOPs) when the hardware implementation is targeted. This affects mainly the energy consumption and time latency.

In this work, a novel touch modality classification framework using Recurrent Neural Networks (RNNs) is proposed. RNNs are mainly designed to deal with sequence prediction problems. They have been very successful in processing natural language, i.e., working on sequences of texts and spoken language that are represented as time series [6], [7]. Data acquired from tactile sensors have 3-dimension tensor structure (similar to a video) where the first two dimensions are defined by the geometry of the sensor array while time defines the third dimension. Hence, given this representation of tactile data, in

M. Alameh, Y. Abbass, G. Moser and M. Valle are with the Department of Electrical, Electronic, Telecommunications Engineering, and Naval Architecture (DITEN) - University of Genoa, via Opera Pia 11A, 16145, Genova, Italy. (e-mail: mohamad.alameh@edu.unige.it, yahya.abbass@edu.unige.it, gabriele.moser@unige.it, maurizio.valle@unige.it)

A. Ibrahim is with the department of Electrical and Electronics Engineering - Lebanese International University, Beirut 1105, Lebanon and with the Department of Electric, Electronic, Telecommunication Engineering and Naval Architecture, University of Genoa, via Opera Pia 11A, 16145, Genova, Italy. (e-mail: ali.ibrahim@edu.unige.it)

this paper we adopt RNNs as they are effective in processing data time series.

The main contributions of this work are summarized as follows:

- We explore the potential of RNN models for touch modality classification. For this purpose, we specifically propose two methods that are based on two separate RNN architectures, namely Long Short Term Memory (LSTM) [8] and Gated Recurrent Unit (GRU) [9] networks to capture long-term dependence from tactile data.

- We propose a case-specific approach to dataset organization to address the peculiarities of tactile data within the aforementioned architectures. For both LSTM and GRU models, averaging with overlap is applied to the input tensor aiming to maintain data about previous time-step in the current time-step.

- The proposed RNN framework for tactile data classification has been experimentally validated with a real dataset and compared to the state of the art achievements. The experimental results demonstrate that the proposed approach achieves a reduction in the number of FLOPs by 99.98% and by 98.34% in memory storage compared to the same problem in the state of the art [10]. Moreover, The computing architecture is scalable, i.e. the complexity is still acceptable when the system is scaled up in terms of input matrix size and number of classes to be recognized. Nonetheless, the classification has reached an accuracy of 84.23% on a 3-class touch modality data set (explained in Section IV), which is higher than state of art solutions on the same dataset [11], [12].

The rest of the paper is organized as follows. Section II reviews the related works in tactile data processing. Section III gives a brief discussion on the methodology followed in the proposed approach. Section IV introduces the experimental setup with the details of the different proposed models. In Section V, we report and analyze the experimental results with a discussion. Finally, we conclude the paper in Section VI.

II. State of Art

Different works in the literature have addressed tactile data processing using machine learning and deep learning, including the use of LSTM networks. In general, these works address both static and dynamic tactile data perception. In static perception, a single reading is captured for a sensor array, while the dynamic aspect deals with tensorial data collected as frames; each frame corresponds to readings at a single time, and the whole tensor is considered as a sample to be processed. In [13] and [14], nine static touch modalities (tap, pat, push, stroke, scratch, slap, pull, squeeze, and no-touch) are classified using LogitBoost [15] and SVM respectively. LogitBoost in [13] degrades in performance when trained on data collected by 40 participants (71%), while SVM achieves an accuracy range of 80.10% - 81.85% in [14].

In [16] two approaches are used to classify eight objects, i.e., finger, hand, arm, pen, scissors, pliers, sticky tape, and Allen-key, using a 28×50 tactile sensory array attached to a robotic arm to collect static pressure maps. For feature extraction, the first approach uses Speeded-Up Robust Features (SURF) descriptor [17], while the second uses a pre-trained AlexNet CNN [18]. Finally, a Support Vector Machine (SVM) [19] classifier is employed for both approaches. In [20], authors use a shallow CNN (only three convolutional layers inside) based on AlexNet to identify 22 objects using static pressure maps, collected from a 28×50 tactile sensory array. While in [21], an optimised embedded implementation of the latter solution is achieved on various hardware platforms.

Rouhafzay et al. [22] employ a combination of virtual tactile sensors (32×32) and visual guidance to distinguish eight classes of simulated objects. Two neural networks are used: a 3D ConvNet for the series of object images coming from tactile sensors and a 1D ConvNet for the series of the normal vectors to the object surface. Abderrahmane et al. [23] introduce a zero-shot object recognition framework, to identify previously unknown objects based on haptic feedback using BioTac sensors [24]. Two CNNs are employed: one for visual data and another for tactile data where the time dimension in the data is transformed into spatial dimension. Gastaldo et al. [11] propose tensor-SVM and tensor-RLS to classify three touch modalities collected as tensorial data from a sensory array. In [12] the same problem is solved using Deep Convolutional Neural Network (DCNN) and transfer learning. In the latter, tensorial sensory data is transformed into synthetic RGB images, and then pre-trained CNN models on ImageNet [25] is used for feature extraction.

In [26], authors collect frames of pressure maps from squeezing an object in contact with a TekScan sensor. 3D CNN are compared to 2D CNN in order to classify three different datasets achieving a higher accuracy when 3D CNN have been employed. Various other works on tactile data processing can be found in [27], [28], and [29] as well.

LSTM networks have recently attracted attention in tactile data processing, especially for the case of data presented in time-series, i.e., each sensor acquires time series of readings, defined by the data readout frequency. In [30], an LSTM is used to predict shape independent hardness of objects from data generated as video from a GelSight sensor with a grid of 960 × 720 pixels. Features from five video frames are extracted using a CNN and then used as an input for an LSTM network.

In [31], a CNN is used for active tactile clothing perception. Color RGB pressure maps generated from a large tactile sensor attached to a robotic arm grasping clothes, are used to classify different textile properties: thickness, smoothness, textile type, washing method, softness, stretchiness, durability, woolen, and wind-proof. Static and dynamic perception are explored, different CNN models are experimented for single image classification, the best performing being the VGG-19 pretrained on ImageNet [25], while a CNN+LSTM model is used for dynamic data, the LSTM is composed has hidden state of dimension 2048.

In [32], authors use an LSTM for slipping prediction over six different material surfaces, using three different tactile sensors, attached to three fingers of a robotic arm. A 20-neuron single-layer LSTM is used in this work. In [33],

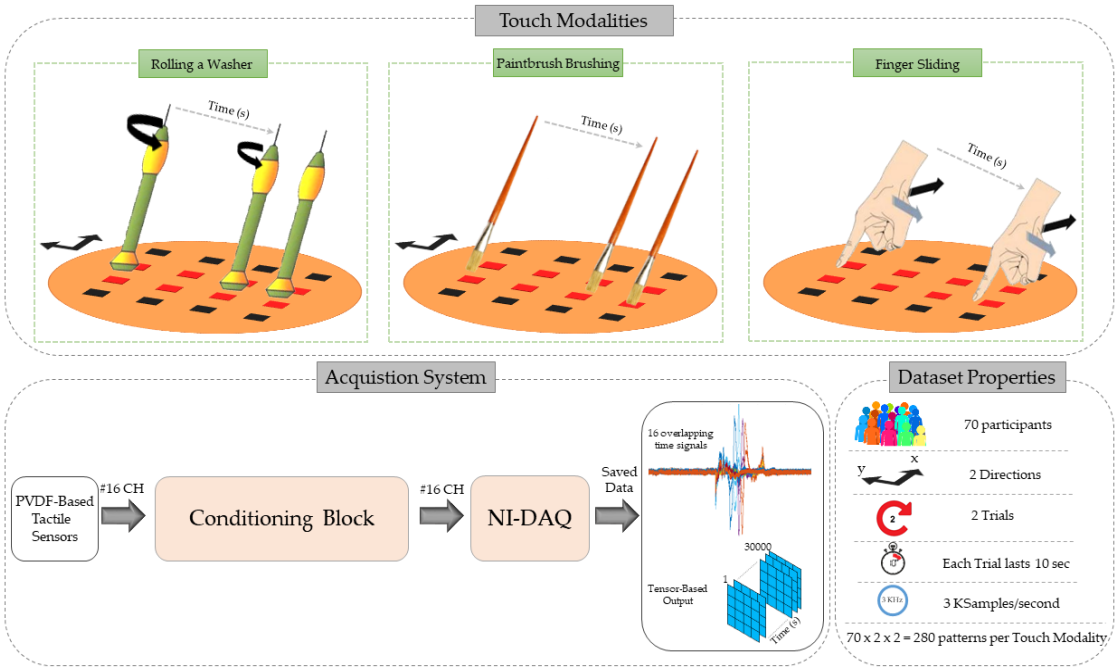


Fig. 1. Scheme of touch modalities, Tactile acquisition system and Dataset Properties.

authors use a CNN and a Graph Convolutional Network [34] for binary grasp stability detection, and an LSTM and a ConvLSTM for detecting slipping direction (translational and rotational). The number of LSTM units is not mentioned for LSTM, instead for ConvLSTM, 5 ConvLSTM layers were used, then pooling, and two fully connected layers, the input for ConvLSTM is 11×12 RGB image. In [35], Dong et al. use high resolution tactile sensor (GelSlim) to control the insertion of objects in a box-packing scenario, they employ two distinct models based on CNN+LSTM, the first network for detecting the direction of the error, and the other for detecting the magnitude of the error, the CNN model used to extract features is a pretrained AlexNet model. The LSTM contains 170 units.

The aforementioned works generally have a high complexity in the learning and inference phases, especially when deep learning is used, which raises challenges for hardware implementation requirements [36]. Here arises the need of low-complexity but high-accuracy touch modality classification solutions suitable for the embedded hardware implementation, where resources are usually limited (e.g., power, memory). In this respect, LSTM is a promising candidate, especially when shared weights are used.
To the best of our knowledge, no works were done, yet, to explore the potential of LSTM for solving the touch modality classification problem.

III. Methodology

According to the state-of-art, different methods were used to address tactile data processing and touch modality classification. In this framework, the capability of deep learning architectures to extract meaningful data representations from high-dimensional spatio-temporal data, without the need for handcrafting features, conveys an especially promising potential. Here, we leverage on this potential to propose two novel methods for tactile data classification, based on recurrent architectures.

On one hand, the planar topology of a tactile array may generally prompt the use of CNN architectures. This strategy is expected to be promising especially if large-area arrays, made of many individual sensors and acquiring tactile imagery with relatively high resolution, are used. On the other hand, highly effective CNN architectures may include a large set of parameters, which creates a challenge for realtime processing, power consumption, and training set size.

Touch modality data have relevant spatio-temporal characteristics: touch occurs in time (temporal aspect) and takes place on the surface of the tactile sensory array (spatial aspect). Using CNN in [12] necessitates to transform temporal characteristics into spatial characteristics, by generating a single synthetic image for each tensorial sample. Instead, RNN intrinsic capabilities to capture time-dependent behaviors make them a promising approach for the analysis of such data. This is achieved by making a recursive input into the network, which comes from the output at the previous time-step. Another important consequence of using RNNs is that the weights are shared across time, i.e., weights are defined for a single RNN block, and these weights will be shared upon execution over all time-steps. This means a reduction in the number of stored trained parameters and in the complexity as well.
Here, the key idea to leverage on these properties of the RNN approach for the classification of touch modalities is explored, and two RNN models (LSTM and GRU) are proposed in this framework. The description of each architecture will follow in the next subsections.

A. LSTM network

LSTM networks [8] are RNNs capable of modeling long-range temporal dependencies [37]. RNNs are composed of a chain of units whose output is connected not only to the next layer but also fed back to the unit itself as an input, thus allowing the information to persist. LSTM behaves in a temporal manner that is appropriate for learning sequential models [38]. A clear example of LSTM usage is the prediction of the next word in a sentence, having observed the previous words [39]. Moreover, LSTM can act as a classifier for time series of data [40], [41]. A key characteristic of LSTM is its memory cell, which acts as an accumulator of the state information. Several self-parameterized controlling gates are used to access, write, and clear the cell (output, input, and forget gates). Details can be found in [39].

B. GRU network

Similar to the LSTM unit, the GRU unit has gates that modulate the flow of information inside the unit [9]. A typical GRU cell is composed of only two gates, the reset gate (whose role is similar to the forget gate of the LSTM) and the update gate (whose role loosely matches the input gate of the LSTM). Details can be found in [9].

C. LSTM and GRU for touch modality classification

Based on the general architecture of the LSTM unit, three parameters are required to build an LSTM network: feature vector length, time-steps, and the number of neurons. The same applies to GRU as well. For these two RNN models, each input pattern is a feature vector representing a sampled time signal. In this perspective, raw data should be pre-processed in order to be suitable for RNN models. The preprocessing has three main objectives: 1) to reduce the input data size to simplify the training, 2) to normalise the data as a general consideration in training neural networks, and 3) to make the dataset format compatible with the network's input format.
The feature vector length for both LSTM and GRU was chosen equal to the size of the tactile array (i.e. 16 in our experiments), therefore each sensor in the tactile array is considered as a feature. The number of neurons was selected on a trial basis, in order to have the fewest trainable parameters as possible, while achieving an acceptable accuracy with respect to the related state of the art. Detailed description of each model and of the related architectures is presented in Section IV.B.

IV. Experimental Setup

The dataset collected in [11] has been considered in this paper, this dataset can be downloaded from: `https://data.mendeley.com/datasets/dmcdp33ctt/2`. Figure 1 illustrates the dataset used in this work. A tactile aquistion system (Figure 1) that is based on a charge amplifier and a Data Acquisition board (DAQ) was used to collect the data from the sensory array. Seventy subjects were asked to perform predetermined touch modalities i.e. sliding the finger, brushing a paintbrush, and rolling a washer. Each participant touches the top surface of a 4×4 piezoelectric tactile sensory array in two moving directions – horizontal and vertical over a random position –, repeating this twice for each position. For every single touch, a 10 seconds acquisition was done at 3 kSamples/second. The collected data were arranged into a 3-dimensional tensor, whose size was tactile sensory array size × number of acquired samples = 4×4×30000. 280 patterns per touch modality in a total of 840 patterns were available.

A. Dataset Organisation

In order to use RNNs for this dataset, preprocessing was applied to the dataset. The first step was to generate a 3D tensor that contains only the useful touch information from the original raw data. In other words, in the first step, we selected the time period where touch is applied as shown in Figure 2.A. This was done by checking at which time instant T any of the sensor output value exceeds a predefined threshold. The resulting time T indicates the starting point of the useful touch data. To have a constant number of frames over all the patterns, we fixed the size of the data to 6144 samples per sensor, i.e., in the $[T, T + 6143]$ range. The average activity time duration for all users is around 2 seconds i.e., 6000 samples. We selected 6144 samples per sensor i.e., 6144 × 16 to make the tensor size a multiple of 64×64, which makes it suitable for comparison with the CNN-based networks that will be mentioned later in section IV.B.

The result from the first step is a 3D tensor of size 4×4×6144 each, having the same original frequency of 3 kSamples/second, and starting at time instant T until reaching 6144 frames. Sixty patterns out of 840 were excluded since no sensor readings exceeded the activity threshold in these patterns. The refined dataset referred later on as Dataset (A), is composed of D tensors of size 4×4×6144 per touch modality (three touch modalities), where D is the number of patterns ($D = 260$). Then, from Dataset (A) two different datasets that fit the models used for experiments are derived.

Dataset (C) is composed of eight different subsamplings of Dataset (A), as shown in Figure 2.C. An averaging with 50%

Algorithm 1: Generate Dataset (C)

Input: Dataset (A), Output: Dataset (C)
a = single pattern from Dataset (A);
K = number of frames in a;
C = output pattern;
F = number of sensors;
N = number of output frames;
$slot \leftarrow K/N$;
for $i \leftarrow 1$ to F **do**
  $sen \leftarrow a(1 : end, i)$; // sen: single sensor output
  $u \leftarrow 1$;
  for $j \leftarrow 1$ to N **do**
    if $u < N$ **then**
      $C(j, i) \leftarrow Average(sen(u : slot + u))$;
    if $u = N$ **then**
      $C(j, i) \leftarrow Average(sen(u : end))$;
    $u \leftarrow u + slot/2$;

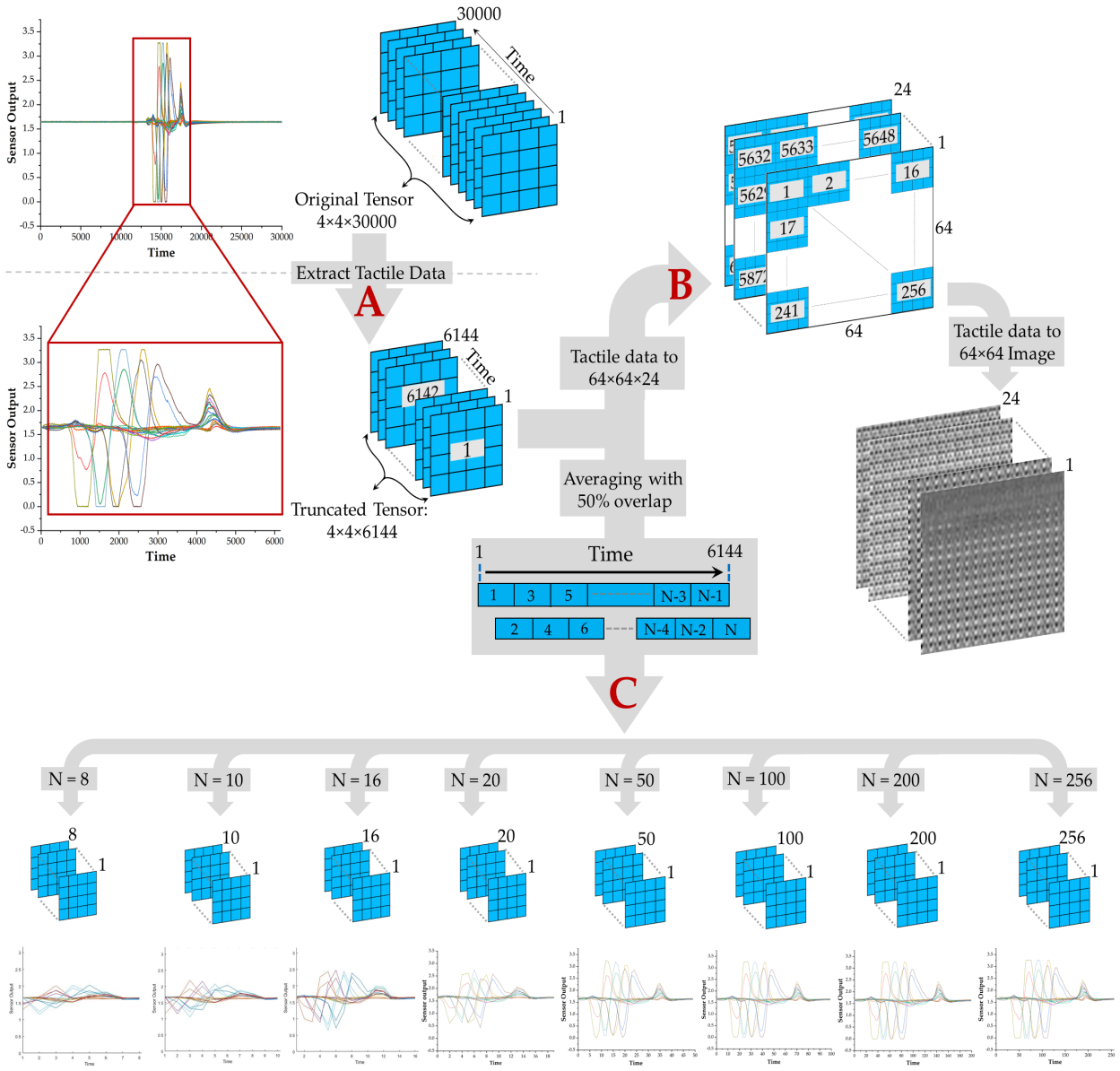


Fig. 2. Dataset Organisation

overlap was used to downsample the tensor from 4×4×6144 to 4×4×N, where N represents the number of the input time-steps for the LSTM and GRU networks. N varies in the following set of values 8, 10, 16, 20, 50, 100, 200, and 256. The overlapping helps maintaining data about the previous time-step in the current time-step. The increment of (slot/2) in the subsampling algorithm clarifies this overlapping. The whole process is described in Algorithm 1. The choice of N in this range, is to detect the subsampling range yielding to an acceptable accuracy, and to make a fair comparison with state-of-art solutions [10] [11] where $N = 20$ is used. For a further comparison, two CNN-based models namely ConvLSTM and CNN-LSTM were also implemented and tested. The reason behind using these two CNN-based models for comparison purposes is due to the ability of CNN of extracting features from images, which was also proved for synthetic images generated from tactile sensors [12], and is aimed at discussing experimentally the choice of adopting an RNN rather than a CNN approach within the proposed approach.

CNNs use convolutional layers that act on data arranged on planar (or possibly multidimensional) grids. Following this characteristic, each pattern in Dataset (A) of size 4×4×6144 was transformed into a time series of larger images, to which convolution operators with non-negligible filter size could be applied. Each 4×4×6144 pattern is rearranged into 64×64×24 samples (image size × time-steps). In order to obtain a feature vector that reflects the time order of the patterns, the 4×4 patterns are arranged in each image following the way the convolutional filters slide along the image [42]. Hence, the time order of the 4×4 blocks influences the output feature vector. Therefore, the time sequence order is maintained within each 64×64 image, which will be called (im_{64}) in the

following. Each im_{64} consists of 256 images of 4×4 pixels. The resulted dataset is called Dataset (B).

The three datasets are shown in Figure 2, where one pattern was used as an example to illustrate the difference in the presentation of the original dataset. All codes used in this article can be downloaded from: `https://hgithub.com/alamehm/IEEE_SENSORS_Touch_modality_/releases/tag/1.0`

B. Implementation

The two architectures framed within the proposed RNN approach to touch data classification and mentioned in Section III,were implemented as described in the next subsections. For each model a dense layer was added at the end for the classification.

1) LSTM network: the LSTM network was composed of one LSTM layer (10 neurons) and a flat input layer of length 16, was trained on Dataset (C). As for the time-steps, the LSTM network was trained for each of the time-step configurations i.e., $N \in \{8, 10, 16, 20, 50, 100, 200, 256\}$.

2) GRU network: The GRU network was composed of a single GRU layer, applied within two alternatives: 10 neurons and 12 neurons per GRU layer. This GRU network also had a flat input layer of length 16 and was separately applied with each aforementioned value of N.

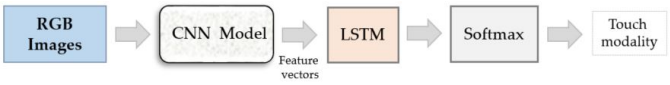


Fig. 3. CNN-LSTM structure.

3) CNN-LSTM network: Four different pre-trained CNN models (ResNet50 [43], ResNet150V2 [44], MobileNetV2 [45] and VGG16 [46]) were considered for comparison purposes to play the role of the CNN model shown in Figure 3. The four CNN models were pre-trained using the ImageNet dataset [25]. These models were selected based on the results of our previous work [12], which showed that the four models were effective in extracting features from the considered dataset. The models were used to transform each image in Dataset (B) into a fixed-length feature vector. Therefore, each CNN model transforms Dataset (B) into a 4D tensor of shape $(D \times 24 \times 1 \times K)$, where D is the number of patterns, K is the size of the output feature vector in each model, and 24 is the time-steps in each pattern. Thus, the input of the LSTM block has K features, and 24 time-steps.

4) ConvLSTM network: The second CNN-based model used for comparison was composed of single ConvLSTM layer (32 filters of size 3 × 3) followed by one fully-connected layer with 100 units and ReLU activation. Dataset (B) was used, the input of the network is a im_{64}, and the time-steps are 24.

C. Training

The training and application of the proposed networks were done using Keras with Tensorflow backend on an NVIDIA GPU. Dataset (C) was normalized before being used to train and test the LSTM and GRU models.

For training / testing split, an 80/20 percentage was chosen. Five folds were generated, in a way that the intersection of testing samples was empty across all folds. The Adam optimiser [47] was used to train the networks, with categorical cross entropy as a loss function and softmax activation for the output layer. Different runs were made also for each fold, including different batch sizes and epochs, in order to find the hyper-parameters that lead to better accuracy. Finally the choice was limited to batch-size=[48,69] and epochs=[48,96], such that we obtained 4 combinations in total. For each combination, ten training runs with random initialization and random batch selection have been made i.e., the batch size is fixed, but choosing the samples for a batch is random. Therefore, for each model mentioned in IV.B, 4×10×5 (combinations×runs×folds) training runs have been made. Finally the accuracy was obtained by averaging all runs across a fold for each combination of (batch-size, epochs). Results in the next section corresponds to the best (batch-size, epochs) combination, i.e., the combination that gave the highest accuracy, in our case it is batch-size=48 and epochs=96. Figure. 4 shows the accuracy obtained on each fold, for the selected (batch-size, epochs) combination, using the four different models.

V. Results and Discussion

The proposed architectures applied by subsampling into N samples for each pattern will be referred to as LSTMN and GRUN in the following. According to Figure 4.a, the use of LSTM20 yielded a higher accuracy than the other subsampling alternatives, and as compared to other tested models. LSTM20 achieved the highest accuracy, i.e., 84.23% with 1113 trainable parameters, and 2950 FLOPs per LSTM block as shown in Table I. Which means a 99.989% reduction in FLOPs and 98.34% reduction in model size, compared to [10] where the same input tensor size of $4 \times 4 \times 20$ has been used.

On the contrary, if we consider the average accuracy, we can see that LSTM8 had the best average, among the various LSTM and GRU configurations, as well as compared to state of art solutions applied on the same dataset [11] [12], as shown in Table I.

Regarding the GRU, it also proved a high accuracy with a smaller number of parameters and a comparable number of FLOPs with respect to LSTM20 as shown in Figure 4.b,c. The 10-neuron GRU8 achieved a best accuracy of 82.11% with 843 trainable parameters and 2228 FLOPs. While a 12-neuron GRU20, with 1083 trainable parameters, and 2960 FLOPs per single GRU block achieved 83.78% as shown in Figure 5 and Table I. GRU8 networks (10 neurons and 12 neurons) achieved the highest average accuracy (73.10% and 73.43% respectively) compared to the other GRU configurations.

Both the GRU and LSTM models have reached an accuracy (83.78% and 84.23%) higher than the best accuracy achieved by state-of-the-art approaches applied to the same dataset, whether in tensor-SVM (76.6%) and tensor-RLS (77.3%) [11], or using DCNN (76.9%) [12].

The Figs 4 (a), (b), and (c) show that when the value of N is approximately into the range 8 - 20, the results are quite stable

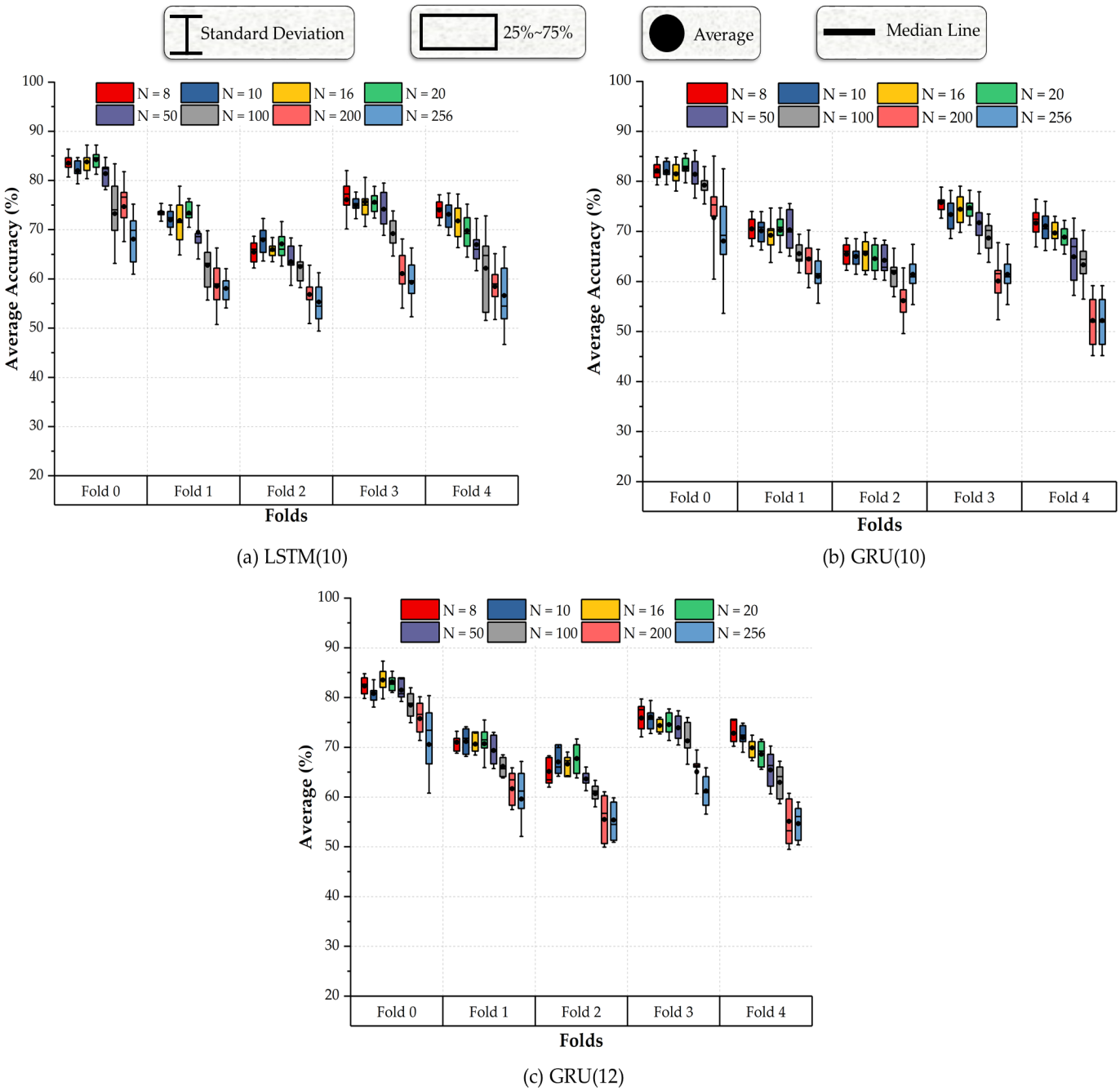


Fig. 4. Average Accuracy achieved varying the input for five different folds: (a) LSTM network (10 neurons) using Dataset C; (b) GRU network (10 neurons) using Dataset C; (c) GRU network (12 neurons) using Dataset C.

(the values of the average accuracy are quite similar). Any value of N in the above-mentioned range provides similar results to the ones obtained with $N = 20$. A consequence is that, the choice of the value of N is not critical if N belongs to the cited range. In some cases, values of N other than 20 seem to provide higher averages in percentage accuracy (e.g., GRU8 (10 neurons), fold 0).

As per ConvLSTM and CNN-LSTM, these models did not converge well, and the obtained accuracies did not exceed 60%. On one hand, this is an expected result, because of the large number of trainable parameters compared to the small dataset size. On the other hand, these results confirm the methodological choice of RNN models for the addressed problem of touch data classification, especially in the addressed case study of a rather small-size tactile array. Notwithstanding that, the previous results in DCNN [12] were more accurate, but that result was accomplished using transfer learning. Specifically in [12], all the used networks (except the final softmax layer used for classification), were pre-trained on millions of images from ImageNet [25] and thousands of classes. Then, the last layer was trained on the considered dataset. On the contrary, in the case of CNN-LSTM we are using pretrained CNNs to extract features from Dataset (B).

TABLE I
COMPARISON OF ACCURACY, NUMBER OF PARAMETERS, AND FLOPS

Model	Fold's Average Accuracy over 10 runs (%)					Best Accuracy (%)	Average Accuracy ± Stdev (%)	Model parameters	FLOPs
	Fold 0	Fold 1	Fold 2	Fold 3	Fold 4				
LSTM8 (10 neurons)	83.52	73.52	65.44	76.08	73.97	83.52	**74.51** ± 6.46	1113	2950×8
LSTM10 (10 neurons)	81.98	71.98	67.94	74.93	73.14	81.98	74 ± 5.14	1113	2950×10
LSTM16 (10 neurons)	83.78	71.85	65.96	75.64	71.79	83.78	73.8 ± 6.56	1113	2950×16
LSTM20 (10 neurons)	84.23	73.39	67.11	75.57	69.80	**84.23**	74.02 ± 6.56	1113	2950×20
LSTM50 (10 neurons)	79.48	69.48	63.52	74.16	66.98	79.48	70.72 ± 6.24	1113	2950×50
LSTM100 (10 neurons)	73.26	62.75	62.5	69.23	62.17	73.26	65.98 ± 5.01	1113	2950×100
LSTM200 (10 neurons)	74.67	58.52	56.85	61.08	58.46	74.67	61.92 ± 7.29	1113	2950×200
LSTM256 (10 neurons)	68.07	58.07	55.32	59.29	56.6	68.07	59.47 ± 5.03	1113	2950×256
GRU8 (10 neurons)	82.11	70.5	65.44	75.76	71.66	**82.11**	**73.10** ± 6.23	**843**	2228×8
GRU10 (10 neurons)	81.98	70.12	65	73.39	71.08	81.98	72.32 ± 6.21	843	2228×10
GRU16 (10 neurons)	81.47	69.23	65.57	74.42	69.67	81.47	72.07 ± 6.12	843	2228×16
GRU20 (10 neurons)	81.92	70.25	64.55	74.74	68.84	81.92	72.06 ± 6.60	843	2228×20
GRU50 (10 neurons)	81.41	70.32	64.23	71.73	64.93	81.41	70.52 ± 6.9	843	2228×50
GRU100 (10 neurons)	79.23	65.57	61.79	68.65	63.33	79.23	67.71 ± 6.93	843	2228×100
GRU200 (10 neurons)	72.75	64.48	56.15	60.06	52.17	72.75	61.12 ± 7.94	843	2228×200
GRU256 (10 neurons)	68.07	61.02	61.41	61.41	52.17	68.07	60.82 ± 5.65	843	2228×256
GRU8 (12 neurons)	82.30	71.02	65.12	75.89	72.82	82.30	**73.43** ± 6.32	1083	2960×8
GRU10 (12 neurons)	80.83	71.15	67.05	76.08	71.92	80.83	73.41 ± 5.24	1083	2960×10
GRU16 (12 neurons)	83.52	70.64	66.66	74.35	69.87	83.52	73.01 ± 6.48	1083	2960×16
GRU20 (12 neurons)	83.78	70.70	67.75	74.55	68.58	**83.78**	73.07 ± 6.53	1083	2960×20
GRU50 (12 neurons)	81.47	69.35	63.65	73.91	65.44	81.47	70.76 ± 7.16	1083	2960×50
GRU100 (12 neurons)	78.46	66.15	60.70	71.28	62.94	78.46	67.91 ± 7.11	1083	2960×100
GRU200 (12 neurons)	75.76	61.66	55.51	65.06	55.12	75.76	62.62 ± 8.46	1083	2960×200
GRU256 (12 neurons)	70.57	59.61	55.38	61.21	54.67	70.57	60.29 ± 6.37	1083	2960×256
Tensor-SVM [11] [10]	-	-	-	-	-	76.6	71 ± 5.6	67200	545M
Tensor-RLS [11]	-	-	-	-	-	77.3	73.7 ± 3.6	-	-
DCNN (InceptionResNetV2) [12]	-	-	-	-	-	76.9	-	54M	109M

The resulting feature vector may range from 2000 (VGG16) to 8000 (Resnet150v2) features. These features are fed into an LSTM network to be trained from scratch, compared to the 16 features used by the proposed LSTM or GRU formulations. Accordingly, there is a significantly higher number of trainable parameters for both CNN-LSTM and ConvLSTM compared to the proposed GRU and LSTM models, as shown in Table I. Raising the number of LSTM layers and number of neurons did help for ConvLSTM and CNN-LSTM,still without reaching a comparable accuracy.

LSTM achieved many benefits: 1) FLOPs and memory occupation i.e., number of trainable parameters are less, compared to the state of the art.

2) As for the computation, data can be fed into an LSTM network, as soon as all input features are ready for a single time-step, i.e., when data are ready at time t, they can be forwarded into an LSTM block, without waiting for the data from all time-steps. Also, data occuring at time $t+1$ can be collected in parallel with respect to the execution of LSTM block of data at time t, as illustrated in Figure 6. While in other solutions like CNN or tensor-SVM, all data should be assembled in order to be able to process them.

3) Higher accuracy is obtained on the considered dataset.

4) The model is highly scalable and independent on the size of the dataset, in terms of both FLOPs and number of trainable parameters, as illustrated in Figure 7. Unlike SVM, where the model size depends on the number of training samples and does not support multi-class labeling directly [48]. In addition, since LSTM used shared weights, e.g., when training an LSTM of $N = 20$ time-steps, weights are shared across all time-steps, which makes a model trained on "N" time-steps data still usable for "M" time-steps data.

VI. CONCLUSION

In this paper, we have investigated the potential of RNNs for touch modality classification. Two different approaches

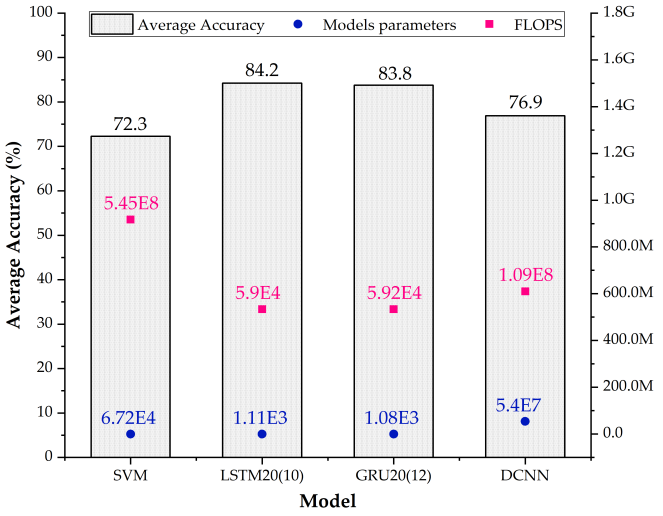


Fig. 5. Accuracy of the best performing networks. LSTM20(10) stands for LSTM network with 10 neurons and $N = 20$. GRU20(12) stands for GRU network with 12 neurons and $N = 20$.

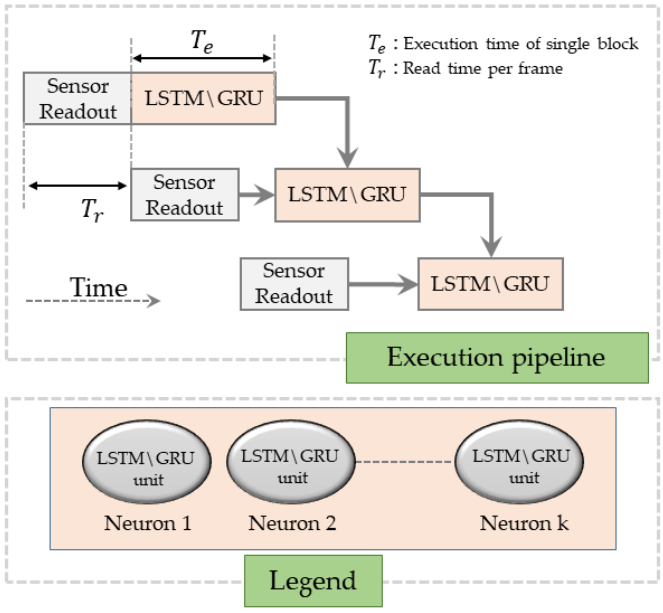


Fig. 6. LSTM\GRU Execution pipeline

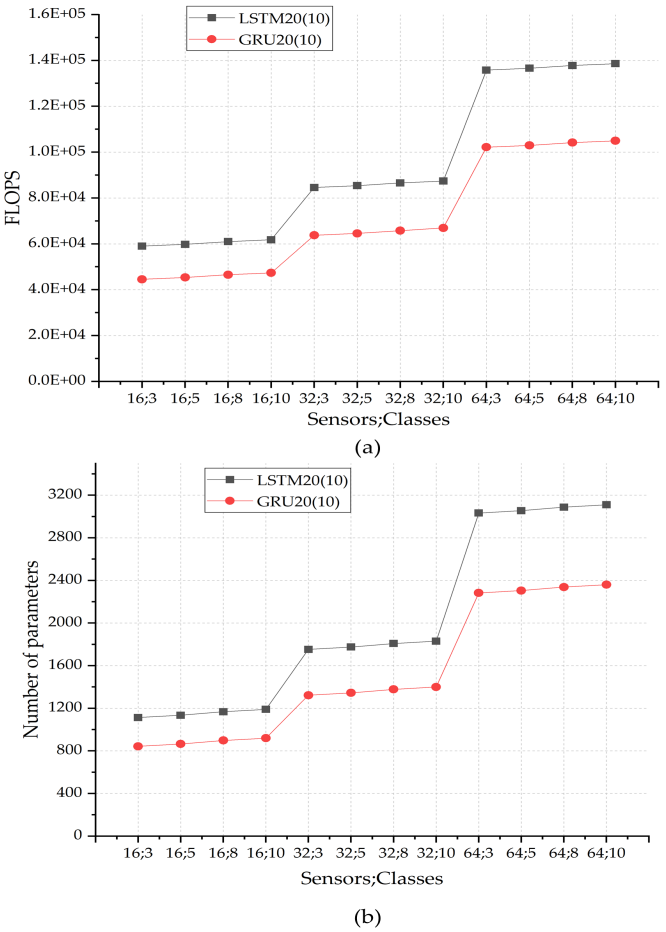


Fig. 7. (a) Number of sensors and classes versus number of FLOPS. (b) Number of sensors and classes versus Model parameters.

based on LSTM and GRU architectures have been proposed to extract long-term dependence from tactile data. Proposed approaches have been validated on real tactile data acquired from 4×4 piezoelectric tactile arrays. Experimental results have shown that the highest achieved accuracy is of 84.23% and 83.78% for LSTM and GRU respectively, compared with a value of 77.3% and 76.9% for the highest accuracy achieved in the literature on the same benchmark dataset (see [11], [12], respectively); Moreover, the average accuracy achieved by LSTM8, is higher than all reported average accuracies, as seen in Table I. On the other hand, the proposed architectures reduces drastically the number of FLOPs considered as the main factor affecting the hardware complexity of the system. The number of FLOPs has been reduced of 99.989% compared to the same problem in the state of the art [10], which will have the impact on time latency, hardware resources, memory storage and energy consumption when the hardware implementation will be targeted. Another important aspect offered by the proposed approach is the scalability in the computing architecture. This means that the complexity of the system remains acceptable when the system is scaled up in terms of input matrix size and number of classes to be recognized which was a main drawback limiting similar state of art solutions [48]. As a conclusion, the proposed work represents a good candidate to be embedded together with a tactile sensing system for robotic or prosthetic applications [1], [2]. Such applications require near-sensor processing with critical constraints such as small hardware area, low energy budget due to the limited battery size, and the low latency needed to perform real-time functions. In future work, we are going to implement the proposed systems on embedded platforms integrated with the tactile sensors and their front-end electronics. Furthermore, after validating its functionality, the system is intended to be implemented on an Application Specific Integrated Circuit (ASIC) to form a smart electronic skin system module for the target applications.

REFERENCES

[1] Ravinder S. Dahiya, Philipp Mittendorfer, Maurizio Valle, Gordon Cheng, and Vladimir J. Lumelsky. Directions Toward Effective Uti-

lization of Tactile Skin: A Review. IEEE Sensors Journal, 13(11):4121–4138, November 2013.

[2] G. Cheng, E. Dean-Leon, F. Bergner, J. R. G. Olvera, Q. Leboutet, and P. Mittendorfer. A Comprehensive Realization of Robot Skin: Sensors, Sensing, Control, and Applications. Proceedings of the IEEE, pages 1–18, 2019.

[3] Ali Ibrahim, Luigi Pinna, Lucia Seminara, and Maurizio Valle. Achievements and Open Issues Toward Embedding Tactile Sensing and Interpretation into Electronic Skin Systems, chapter 23, pages 571–594. John Wiley & Sons, Ltd, 2017.

[4] Marta Franceschi, Lucia Seminara, Strahinja Dosen, Matija Strbac, Maurizio Valle, and Dario Farina. A system for electrotactile feedback using electronic skin and flexible matrix electrodes: Experimental evaluation. IEEE transactions on haptics, 10(2):162–172, 2016.

[5] Ravinder Dahiya, Nivasan Yogeswaran, Fengyuan Liu, Libu Manjakkal, Etienne Burdet, Vincent Hayward, and Henrik Jörntell. Large-area soft e-skin: the challenges beyond sensor designs. Proceedings of the IEEE, 107(10):2016–2033, 2019.

[6] Ji Young Lee and Franck Dernoncourt. Sequential short-text classification with recurrent and convolutional neural networks. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 515–520, San Diego, California, June 2016. Association for Computational Linguistics.

[7] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of cnn and rnn for natural language processing. arXiv preprint arXiv:1702.01923, 2017.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural Computation, 9(8):1735–1780, 1997.

[9] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, pages 1724–1734, 2014.

[10] M. Osta, A. Ibrahim, M. Magno, M. Eggimann, A. Pullini, P. Gastaldo, and M. Valle. An Energy Efficient System for Touch Modality Classification in Electronic Skin Applications. In 2019 IEEE International Symposium on Circuits and Systems (ISCAS), volume 2019-May, pages 1–4. IEEE, may 2019.

[11] Paolo Gastaldo, Luigi Pinna, Lucia Seminara, Maurizio Valle, and Rodolfo Zunino. Computational intelligence techniques for tactile sensing systems. Sensors (Switzerland), 14(6):10952–10976, 2014.

[12] Mohamad Alameh, Ali Ibrahim, Maurizio Valle, and Gabriele Moser. DCNN for Tactile Sensory Data Classification based on Transfer Learning. In 2019 15th Conference on Ph.D Research in Microelectronics and Electronics (PRIME), pages 237–240, July 2019.

[13] David Silvera Tawil, David Rye, and Mari Velonaki. Interpretation of the modality of touch on an artificial arm covered with an eit-based sensitive skin. The International Journal of Robotics Research, 31(13):1627–1641, 2012.

[14] Mohsen Kaboli, Alex Long, and Gordon Cheng. Humanoids learn touch modalities identification via multi-modal robotic skin and robust tactile descriptors. Advanced Robotics, 29(21):1411–1425, 2015.

[15] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The annals of statistics, 28(2):337–407, 2000.

[16] Juan M. Gandarias, Jesus M. Gomez-de Gabriel, and Alfonso Garcia-Cerezo. Human and object recognition with a high-resolution tactile sensor. In 2017 IEEE SENSORS, pages 1–3, Glasgow, October 2017. IEEE.

[17] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In European conference on computer vision, pages 404–417. Springer, 2006.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[19] Vladimir N Vapnik. An overview of statistical learning theory. IEEE transactions on neural networks, 10(5):988–999, 1999.

[20] Juan M. Gandarias, Alfonso J. Garcia-Cerezo, and Jesus M. Gomez-de Gabriel. CNN-Based Methods for Object Recognition With High-Resolution Tactile Sensors. IEEE Sensors Journal, 19(16):6872–6882, 2019.

[21] Mohamad Alameh, Yahya Abbass, Ali Ibrahim, and Maurizio Valle. Smart Tactile Sensing Systems Based on Embedded CNN Implementations. Micromachines, 11(1):103, 2020.

[22] Ghazal Rouhafzay and Ana-Maria Cretu. An Application of Deep Learning to Tactile Data for Object Recognition under Visual Guidance. Sensors, 19(7):1534, January 2019.

[23] Zineb Abderrahmane, Gowrishankar Ganesh, André Crosnier, and Andrea Cherubini. Visuo-Tactile Recognition of Daily-Life Objects Never Seen or Touched Before. In 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), pages 1765–1770. IEEE, 2018.

[24] Chia Hsien Lin, Todd W. Erickson, Jeremy A. Fishel, Nicholas Wettels, and Gerald E. Loeb. Signal processing and fabrication of a biomimetic tactile sensor array with thermal, force and microvibration modalities. In 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO), pages 129–134, December 2009.

[25] ImageNet. http://www.image-net.org. [Online]; Accessed:2019-11-20.

[26] Francisco Pastor, Juan M Gandarias, Alfonso J García-Cerezo, and Jesús M Gómez-de Gabriel. Using 3d convolutional neural networks for tactile object recognition with robotic palpation. Sensors, 19(24):5356, 2019.

[27] Jianhua Li, Siyuan Dong, and Edward Adelson. Slip detection with combined tactile and visual information. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 7772–7777. IEEE, 2018.

[28] Haoying Wu, Daimin Jiang, and Hao Gao. Tactile motion recognition with convolutional neural networks. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1572–1577. IEEE, 2017.

[29] Jennifer Kwiatkowski, Deen Cockburn, and Vincent Duchaine. Grasp stability assessment through the fusion of proprioception and tactile signals using convolutional neural networks. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 286–292, Vancouver, BC, September 2017. IEEE.

[30] Wenzhen Yuan, Chenzhuo Zhu, Andrew Owens, Mandayam A. Srinivasan, and Edward H. Adelson. Shape-independent hardness estimation using deep learning and a GelSight tactile sensor. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 951–958, May 2017.

[31] Wenzhen Yuan, Yuchen Mo, Shaoxiong Wang, and Edward H. Adelson. Active clothing material perception using tactile sensing and deep learning. 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 1–8, 2017.

[32] Karl Van Wyk and Joe Falco. Slip Detection: Analysis and Calibration of Univariate Tactile Signals. arXiv:1806.10451 [cs], June 2018. arXiv: 1806.10451.

[33] Brayan S. Zapata-Impata, Pablo Gil, and Fernando Torres. Tactile-Driven Grasp Stability and Slip Prediction. Robotics, 8(4):85, 2019.

[34] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.

[35] Siyuan Dong and Alberto Rodríguez. Tactile-based insertion for dense box-packing. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 7953–7960, 2019.

[36] V. Sze, Y. Chen, J. Emer, A. Suleiman, and Z. Zhang. Hardware for machine learning: Challenges and opportunities. In 2017 IEEE Custom Integrated Circuits Conference (CICC), pages 1–8, 2017.

[37] Xingjian Shi, Zhourong Chen, Hao Wang, Dit Yan Yeung, Wai Kin Wong, and Wang Chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in Neural Information Processing Systems, 2015-Janua:802–810, 2015.

[38] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. LSTM Fully Convolutional Networks for Time Series Classification. IEEE Access, 6:1662–1669, 2017.

[39] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In Thirteenth annual conference of the international speech communication association, 2012.

[40] Sreeraj Rajendran, Wannes Meert, Domenico Giustiniano, Vincent Lenders, and Sofie Pollin. Deep learning models for wireless signal classification with distributed low-cost spectrum sensors. IEEE Transactions on Cognitive Communications and Networking, 4(3):433–445, 2018.

[41] Guangyi Zhang, Vandad Davoodnia, Alireza Sepas-Moghaddam, Yaoxue Zhang, and Ali Etemad. Classification of hand movements from eeg using a deep attention-based lstm network. IEEE Sensors Journal, 2019.

[42] Anirudha Ghosh, Abu Sufian, Farhana Sultana, Amlan Chakrabarti, and Debashis De. Fundamental Concepts of Convolutional Neural Network. In Intelligent Systems Reference Library, volume 172, pages 519–567. Springer, 2020.

[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. 2016 IEEE Conference on

Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016.
[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. arXiv:1603.05027 [cs], March 2016. arXiv: 1603.05027.
[45] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861 [cs], April 2017. arXiv: 1704.04861.
[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2015.
[47] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
[48] Ali Ibrahim, Paolo Gastaldo, Hussein Chible, and Maurizio Valle. Real-time digital signal processing based on fpgas for electronic skin implementation †. Sensors, 17(3):558, Mar 2017.

Mohamad Alameh (GS'18) Obtained his bachelor degree from CNAM-Paris in "Computer Engineering" in 2008. Started his professional career since 2004, in different computer fields (IT, Software development,etc.). Later on, he obtained his Masters degree in "Information and Decision Support Systems" from Lebanese University - Beirut in 2014. Joined the University of Genoa in 2017, as a Ph.D student in COSMIC Lab, DITEN.

Yahya Abbass (YA) (GS'18) received a B.S. degree in Electronic and an M.Sc. degree in Signal, Telcome, Speech, and Image in 2016 and 2018, respectively, from the Faculty of Science, Lebanese University. From June to August 2018 he was an Internship at ULCO University France. Currently, he is a PhD student in Science and Technology for Electronic and Telecommunication Engineering at the Department of Naval, Electrical, Electronic, and Telecommunications Engineering (DITEN) of the University of Genoa. His main research interest includes Deep Learning, integrated sensing systems, electronic tactile sensors, and feedback systems in prosthetics.

Ali Ibrahim received the M.S. degree in industrial control from the Doctoral School of Sciences and Technologies, Lebanese University, in 2009, and the dual Ph.D. degrees in electronic and computer engineering and robotics and telecommunications from the University of Genova and the Lebanese University in 2016. From 2009 to 2013, he was a Project Designer in electronics in Beirut. He has been a Post-Doctoral Researcher with the Department of Electric, Electronic, Telecommunication Engineering and Naval Architecture, University of Genoa from 2016 to 2018. Currently, he is an assistant professor in the department of Electrical and Electronics Engineering at the Lebanese international University in Lebanon and also an associate researcher Department of Electric, Electronic, Telecommunication Engineering and Naval Architecture, University of Genoa. His research interests involve Embedded machine learning for edge computing, embedded electronic systems, FPGA implementation, digital data processing, and interface electronics for electronic skin systems, approximate computing, and techniques and methods for energy efficient embedded computing.

Gabriele Moser (Senior Member, IEEE) received the Laurea (M.Sc. equivalent) degree in telecommunications engineering and the Ph.D. degree in space sciences and engineering from the University of Genoa, Genoa, Italy, in 2001 and 2005, respectively. Since 2014, he has been an Associate Professor of telecommunications with the University of Genoa, where he has been cooperating with the Image Processing and Pattern Recognition for Remote Sensing Laboratory since 2001 and has been the Head of the Remote Sensing for Environment and Sustainability Laboratory at the Savona Campus since 2013. From January to March 2004, he was a Visiting Student with the Institut National de Recherche en Informatique et en Automatique (INRIA), Sophia Antipolis, France. From 2012 to 2016, he was an External Collaborator of the Ayin Laboratory, INRIA. In 2016, he spent a period as a Visiting Professor with the Institut National Polytechnique de Toulouse, Toulouse, France. His research activity is focused on pattern recognition and image processing methodologies for remote sensing and energy applications. Dr. Moser received the Best Paper Award at the 2010 IEEE Workshop on Hyper spectral Image and Signal Processing and the Interactive Symposium Paper Award at the IEEE International Geoscience and Remote Sensing Symposium (IGARSS) 2016. He served as the Chair of the IEEE GRSS Image Analysis and Data Fusion Technical Committee (IADF TC) from 2013 to 2015 and the IADF TC Co-Chair from 2015 to 2017. He was the Publication Co-Chair of the 2015 IEEE IGARSS, the Technical Program Co-Chair of the IEEE GRSS EARTHVISION Workshop at the 2015 IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR), and a Co-Organizerof the second edition of EARTHVISION at CVPR 2017. He has been an Associate Editor of the IEEE GEOSCIENCE AND REMOTESENSING LETTERS since 2008. He was an Area Editor of PATTERN RECOGNITION LETTERS(PRL) from 2015 to 2018, an Associate Editor of PRL from 2011 to 2015, and a Guest Co-Editor of the September 2015 special issue of the IEEE Geoscience and Remote Sensing Magazine.

Maurizio Valle (MV) (SM'16) received the M.S. degree in Electronic Engineering in 1985 and the Ph.D. degree in Electronics and Computer Science in 1990 from the University of Genova, Italy. In 1992 he joined the University of Genova, first as an assistant and in 2007 as an associate professor. From December 2019, MV is full professor at the Department of Electrical, Electronic and Telecommunications Engineering and Naval Architecture, University of Genova; MV leads the Connected Objects, Smart Materials, Integrated Circuits – COSMIC laboratory. MV has been and is in charge of many research contracts and projects funded at local, national and European levels and by Italian and foreign companies. Professor Valle is co-author of more than 200 papers on international scientific journals and conference proceedings and of the book: Robotic Tactile Sensing Technologies and System, Springer Science+Business Media, Dordrecht, pp. 1–248, 2013 (ISBN: 978-94007-0578-4). His research interests include electronic and microelectronic systems, material integrated sensing systems, tactile sensors and electronic-skin systems. He is IEEE senior member and member of the IEEE CAS Society.