POLITECNICO DI TORINO
Repository ISTITUZIONALE

Enhanced Exploration of Neural Network Models for Indoor Human Monitoring

Original
Enhanced Exploration of Neural Network Models for Indoor Human Monitoring / Subbicini, G., Lavagno, L., Lazarescu, M.T.. - ELETTRONICO. - (2023), pp. 109-114. (2023 9th International Workshop on Advances in Sensors and Interfaces (IWASI) Monopoli (Bari), Italy 08-09 June 2023) [10.1109/IWASI58316.2023.10164436].

Availability:
This version is available at: 11583/2979917 since: 2023-09-06T06:28:03Z

Publisher:
IEEE

Published
DOI:10.1109/IWASI58316.2023.10164436

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

27 November 2026

Enhanced Exploration of Neural Network Models for Indoor Human Monitoring

1st Giorgia Subbicini
Electronics and Telecommunications
Politecnico di Torino, Torino, Italy
e-mail: giorgia.subbicini@polito.it

2rd Luciano Lavagno
Electronics and Telecommunications
Politecnico di Torino, Torino, Italy
e-mail: luciano.lavagno@polito.it

3nd Mihai T. Lazarescu
Electronics and Telecommunications
Politecnico di Torino, Torino, Italy
e-mail: mihai.lazarescu@polito.it

***Abstract*—Indoor human monitoring can enable or enhance a wide range of applications, from medical to security and home or building automation. For effective ubiquitous deployment, the monitoring system should be easy to install and unobtrusive, reliable, low cost, tagless, and privacy-aware. Long-range capacitive sensors are good candidates, but they can be susceptible to environmental electromagnetic noise and require special signal processing. Neural networks (NNs), especially 1D convolutional neural networks (1D-CNNs), excel at extracting information and rejecting noise, but they lose important relationships in max pooling operations. We investigate the performance of NN architectures for time series analysis without this shortcoming, the capsule networks that use dynamic routing, and the temporal convolutional networks (TCNs) that use dilated convolutions to preserve input resolution across layers and extend their receptive field with fewer layers. The networks are optimized for both inference accuracy and resource consumption using two independent state-of-the-art methods, neural architecture search and knowledge distillation. Experimental results show that the TCN architecture performs the best, achieving 12.7 % lower inference loss with 73.3 % less resource consumption than the best 1D-CNN when processing noisy capacitive sensor data for indoor human localization and tracking.**

***Index Terms*—Capacitive sensor, indoor person monitoring, neural network, convolutional neural network, long short-term memory, capsule network, temporal convolutional network, knowledge distillation.**

I. Introduction

Indoor human monitoring is a key enabler for smart environment applications. It can provide insights into how people interact with their environment to help improve safety, security, productivity, energy efficiency, health, and assisted living, especially for the elderly or those with health conditions. For example, sensors can detect falls, track vital signs, or analyze discrepancies in daily routines that may indicate the onset of medical conditions and alert caregivers or emergency services.

Indoor monitoring systems can be broadly classified as active or passive, based on the level of human cooperation expected for effective monitoring. Passive monitoring systems may be more acceptable to the aging population because they are easier to conceal for unobtrusive monitoring, do not require interaction, and do not require frequent maintenance.

Techniques based on the properties of electromagnetic waves, such as Wi-Fi [1], ZigBee [2], and ultra-wide band [3] can cover large areas, but they rely on tag devices. The most popular tagless solutions include sensing of infrared radiation sensing of the human body [4] to generate thermal images, ultrasound sensing [5] based on the time-of-flight of ultrasound signals, and radio detection and ranging [6]. Systems based on images tend to be expensive and raise important privacy concerns. Load mode [7] long-range capacitive sensors [8] use single-plate transducers and the electrically conductive human body as the other plate. They can be inexpensive, relatively easy to conceal aesthetically, and respectful of privacy. However, they are also highly susceptible to environmental noise, which affects the localization accuracy, stability, and sensing range. Their noise immunity can be improved by the design of their sensing plate and electromagnetic field, or by signal acquisition and processing techniques [9].

Long-range single-plate capacitive were used to localize and track persons indoor using neural networks (NNs) [10], concluding that 1D convolutional neural networks (1D-CNNs) have the best accuracy. However, the max pooling operations used in convolutional neural networks (CNNs) are known to lose important relations [11].

We explore two NNs designed for sequential data analysis without max pooling, the temporal convolutional network (TCN) [12]–[14] and the capsule network (CapsNet) [15], [16], and use two state-of-the-art optimization techniques, the neural architecture search (NAS) [17] and the knowledge distillation (KD) [18], [19].

To the best of our knowledge, our main contributions are:

- using temporal convolutional network (TCN) and capsule network (CapsNet) NNs to track human activity indoor with long-range load-mode capacitive sensors;
- using noisy sensor data to compare the performance of TCN and CapsNet NNs, which do not use max pooling, with the 1D-CNN state-of-the-art;
- combining neural architecture search and knowledge distillation techniques to optimize NNs for both resource consumption and accuracy.

II. Related Work

Indoor person localization has been implemented using sensing floor tiles with complex sensors [20], [21], simple electrodes [22], or electromagnetic fields from power lines [23]. Although unobtrusive to human activity and easier to install in existing environments than pressure sensors [24], the system is complex: 9 floor electrodes are used to cover $1.8\,\mathrm{m} \times 1.8\,\mathrm{m}$ [22] or 180 nodes to cover $2.4\,\mathrm{m} \times 2\,\mathrm{m}$ [20].

Machine learning, such as support vector machines or random forests, are often used to handle nonlinear functions and noisy data from Wi-Fi signal fingerprinting for indoor person localization [25]. Ye *et al.* [26] have used CAPSNET for indoor localization using a Wi-Fi network spread over 3 rooms with an average error of 0.68 m, outperforming machine learning based on CNN, support vector machine, CNN with stacked autoencoders, and k-nearest neighbor. Jia *et al.* [27] used TCN for indoor localization with Wi-Fi fingerprints in a reading area of a library (965.6 m^2) with an average error of 3.73 m, outperforming support vector machine, k-nearest neighbor, decision tree, random forest, and an 8-layer multilayer perceptron.

Tariq *et al.* [10] evaluate several NN architectures for indoor person localization and tracking with data collected from a tagless localization system with four single-plate long-range capacitive sensors operating in load mode [28], mounted in the middle of the four sides of a $3\,\mathrm{m} \times 3\,\mathrm{m}$ experimental space. The sensors are noisy and have a pronounced nonlinear characteristic, exposing the NN abilities to denoise, infer, and generalize both the location [26], for which the 1D-CNN excels, and the human motion dynamics, for which the long short-term memory gives the best results.

We use the experimental time series data from the noisy sensors, with non-linear characteristics, and the same preprocessing as [10], but we evaluate two NNs, TCN and CAPSNET, which do not use the max pooling technique of the 1D-CNN, known to discard potentially significant data relationships. For the design space exploration to optimize the performance of NNs, unlike [10], two state-of-the-art techniques, NAS and KD, are used to improve both NN accuracy and resource consumption.

III. BACKGROUND

A. Temporal Convolutional Networks

The state of the art suggests using recurrent and recursive networks for sequence modeling tasks, but they have two major drawbacks: exploding/vanishing gradients and high resource consumption. Recent works combine the low-level spatio-temporal features extraction using CNN with the classification of high-level temporal information using recurrent neural networks ([29]–[31]). Bai *et al.* [12] argue that convolutional networks are best suited for modeling sequential data, obtaining good performance using TCNs. They use multiple layers of exponentially increasing dilated convolutions (holes between two adjacent taps, see Fig. 1) to cover wider ranges of inputs with fewer resources. The convolution blocks are followed by normalization, nonlinear activation, and a dropout layer for regularization, forming residual blocks (two identical sub-blocks of dilated convolutions and a residual connection). The receptive field R_{field} (number of time steps available to filters to predict the element at time step t) is

$$R_{\mathrm{field}} = 1 + 2 \cdot (K_{\mathrm{size}} - 1) \cdot N_{\mathrm{res_block}} \cdot \sum_i d_i \qquad (1)$$

where $N_{\mathrm{res_block}}$ is the number of stacked residual blocks, d is a vector containing the dilations along the hidden layers, and K_{size} is the kernel size.

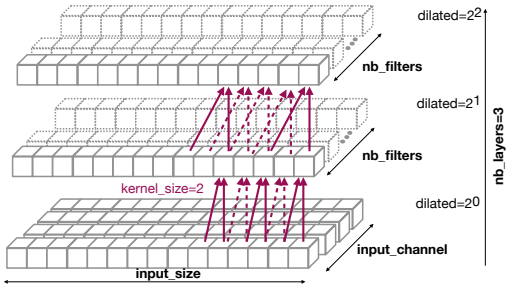

Fig. 1. Dilated causal convolutional blocks of a temporal convolutional network (TCN) have an input tensor of `input_size` length repeated `input_channel` times. Each dilated convolution block has `nb_layers`, each with `nb_filters` of `kernel_size` (purple arrows). The dilation factor (hole size between convolved elements) increases exponentially along the hidden layers.

As Tariq *et al.* [10], we use a bidirectional TCN (which infers based on both past and future tuples) with an input window of 5 s (15 tuples). The TCN R_{field} should be equal or larger than the input window, which can be processed with a reasonable depth of dilated convolution, in one residual block and a dilation factor of 2, without backpropagation issues. The dilated convolutional block parameters are then refined using NAS, as described in Section II.

B. Capsule Networks

While CNNs can extract sophisticated features with simple computations that are invariant to translations, they (1) fail at rotations and shrinking/enlargement transformations, (2) cannot understand hierarchical and relational structures, and (3) their inference is brittle mainly because of their average/max pooling layers [32], which increase their field of view but may discard relevant features where they are not the maximum or overlook complex patterns that require finer resolution.

CAPSNETS overcome the CNN major limitations due to information loss using dynamic routing instead of max pooling operations. They decompose a complex novel object into a hierarchical representation of previously learned patterns. While a CNN neuron outputs a scalar that signals only if a feature is recognized, without relative object relationships, CAPSNETS explicitly model these relationships in the vector-form of the neuron outputs: the vector length (modulus) encodes the detection probability, while the direction (placement in space) encodes the feature/object state (instantiation parameters e.g., pose, deformation, velocity). The backbone implementation in [16] (see Fig. 2) is composed of two `1D-Convolutional` layers with `ReLU` activation to extract basic features. Then a `PrimaryCaps` layer does feature combination and encapsulation. Each nb_caps1 capsule applies dim_caps1 × kernel_size × nb_filters convolutional kernels. Then `ClassCaps` has nb_class represented as dim_caps2 vectors, with length encoding the detection probability, and direction encoding the state of the recognized class/feature. Finally, we add two fully connected layers which take as input the vector with the highest detection probability and output the predicted (x, y) coordinates.

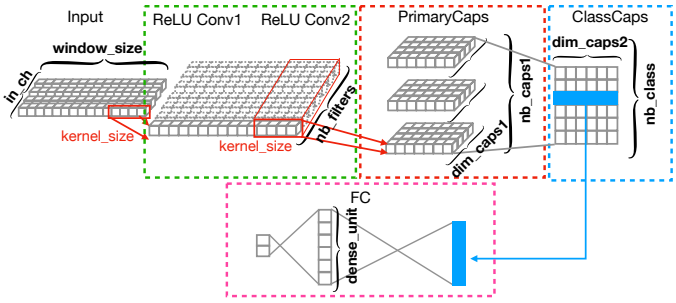

Fig. 2. Capsule network (CapsNet) architecture has two `1D-convolutional` layers, a `PrimaryCaps` layer where encapsulation takes place, a `ClassCaps` high-level feature capsule layer where low-level features converge, and the vector with the highest detection probability is fed to a fully connected layer.

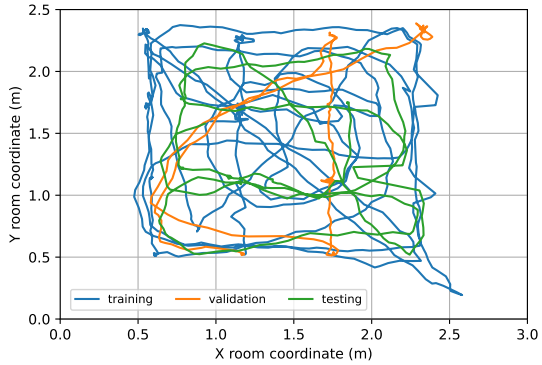

Fig. 3. Full trajectory of the person in the experiment $3\,\mathrm{m} \times 3\,\mathrm{m}$ space divided into segments: 60 % for training, 20 % for validation, and 20 % for testing.

IV. Methodology and Experimental Setup

Our main goal is to improve the performance and the resource consumption of the NNs used for indoor human tracking with low-end noisy capacitive sensors [10].

A. Input Data

To compare the results with Tariq *et al.* [10], the same sensor data and preprocessing are used. The input data come from four single-plate load-mode capacitive sensors mounted in the center of the virtual walls in the $3\,\mathrm{m} \times 3\,\mathrm{m}$ laboratory experimental space, labeled with the person's coordinates, sampled at 3 Hz, and preprocessed using a median filter with a 50 s input window followed by a lowpass filter with a transition band within 0.3–0.4 Hz. The data set is divided into 60 % for training, 20 % for validation, and 20 % for testing, each in time order (see Fig. 3).

B. Neural architecture search

NAS aims to automate the design of NN to a level equal to or better than hand-designed architectures. It can optimize the NN architecture, and estimate or test its performance. NAS is a subfield of automated machine learning, closely related to hyperparameter optimization and meta-learning.

As NAS was used AutoKeras [33], an automated machine learning system based on Keras [34] using a controller for generating NN architectures with a predefined grammar and encoding scheme, a searcher for evaluating the architectures with criteria such as accuracy, complexity, and resource consumption, and a trainer for training and validating the architectures.

All relevant TCN parameters are optimized by NAS (see Fig. 1): the number of dilated convolution filters in each hidden layer of the dilated convolution block, $\mathrm{nb_filters} = [8, 16, 32]$, the number of input tuples to convolve, $\mathrm{kernel_size} = [2, 3, 5]$, the number of hidden layers in the dilated convolution block, $\mathrm{nb_layers} = [2, 3, 5]$, the size of a dense layer before the output, $\mathrm{dense_unit} = [0, 8, 16, 32]$. The number of residual blocks (set to 1 due to the small input window) and the dilation base (set to 2) are not tuned by NAS.

Also for the CapsNet (see Fig. 2), the NAS optimizes all relevant parameters: the number of convolution filters in the layers `1DConv1` and `1DConv2`, $\mathrm{nb_filters} = [8, 16, 32]$, the number of convolved tuples, $\mathrm{kernel_size} = [2, 3, 5]$, the number of capsules in the `PrimaryCaps` layer, $\mathrm{nb_caps1} = [7, 10, 12]$, the dimension of the capsules in the `PrimaryCaps` layer, $\mathrm{dim_caps1} = [3, 5, 7]$, the number of high-level classes in the `ClassCaps` layer, $\mathrm{nb_class} = [3, 5, 7]$, the dimension of the class vectors in the `ClassCaps` layer, $\mathrm{dim_caps2} = [3, 5, 7]$, the size of two dense layers before the output, $\mathrm{dense_unit} = [0, 8, 16, 32]$. The routing iteration is set to 3 and the convolution stride $\mathrm{PrimaryCaps} = 2$ [35]–[37].

The NAS is repeated 3 times for each NN type. The AutoKeras tuner tries 50 different parameter combinations, retraining each 20 times for 800 epochs using the Adamax optimizer tuned autonomously by AutoKeras.

C. Knowledge distillation

Knowledge distillation can compact larger NN models (teachers) into smaller ones (students) through knowledge transfer to reduce resource consumption to simplify deployment on edge devices, improve generalization, and speed up inference.

A TCN and a CapsNet were optimized by NAS to minimize the loss, to be used as teachers, which then transfer their knowledge to smaller student NNs, of the same type, using in turn three of the major KD formulations [19]:

- *Imitation loss regularization* (I_{loss}), implies that the student uses also labels sampled from the teacher distribution, as regularization, in addition to the ground-truth labels.
- *Attentive imitation loss* uses an attention mechanism to measure the similarity between the features of the teacher and the student, and the loss function encourages the student to imitate the teacher behavior.
- *Teacher loss as upper bound*, which implies that the loss function used to train a student to imitate a teacher should not exceed the loss function used to train the teacher itself, which is assumed to be more accurate and reliable. The student and distillation losses are usually weighted to balance their importance.

V. Experimental Results

As Tariq *et al.* [10], the performance of the NNs is evaluated using the mean squared error (MSE), average euclidean distance error (ADE), speed, acceleration (smoothness), and resource consumption (number of parameters).

TABLE I

OUR TCN AND CAPSNET NNS OPTIMIZED USING NEURAL ARCHITECTURE SEARCH (NAS) COMPARED TO [10] (HIGHLIGHTED) THROUGH THE NUMBER OF PARAMETERS, MEAN SQUARED ERROR (MSE), AND AVERAGE EUCLIDEAN DISTANCE ERROR (ADE)

Model	Number of parameters	MSE (m^2)	ADE (m)
1D-CNN (2 layers)	14 530	0.078	0.343
1D-CNN (4 layers)	7618	**0.063**	**0.307**
1D-CNN (6 layers)	8018	0.078	0.328
LSTM (bidirectional)	**2754**	0.079	0.326
CAPSNET (NAS)	5996	**0.063**	**0.303**
TCN (NAS)	**2034**	0.065	0.309

Table I shows the best results of our optimized networks, TCN and CAPSNET, compared to [10] (grey background).

The inference accuracy of the CAPSNET optimized by NAS (MSE $= 0.063\,\mathrm{m}^2$ and ADE $= 0.303\,\mathrm{m}$) matches the best in [10], but consumes less resources (5996 parameters) when it is configured with nb_filters $= 16$, kernel_size $= 2$, nb_caps1 $= 10$, dim_caps1 $= 3$, nb_class $= 3$, dim_caps2 $= 5$, 2 dense layers with dense_unit $= 32$, and trained with Adamax with learning rate 0.001.

The TCN optimized by NAS has inference accuracy (MSE $= 0.065\,\mathrm{m}^2$, ADE $= 0.309\,\mathrm{m}$), slightly higher than the best in [10] ($0.063\,\mathrm{m}^2$), but it achieves these results with much fewer resources (2034 parameters vs. 7618) when configured with nb_filters $= 8$, kernel_size $= 5$, nb_layers $= 3$, dense_unit $= 8$, and trained with Adamax with learning rate 0.0001.

Fig. 4 shows the inference of the X and Y coordinates of the person in the room by the TCN optimized by NAS, the inference of the best 1D-CNN (also the best NN) in [10], and the ground truth for reference. For both coordinates, the TCN inference shows more susceptibility to noise (more oscillations) than the 1D-CNN, and it also seems to infer less accurately the extremes of either coordinate. Fig. 5 allows us to evaluate similarly the CAPSNET optimized by NAS. Its inference appears smoother than the TCN, less afflicted by noise, often comparable and occasionally exceeding the quality of the inference of the 1D-CNN in [10], e.g., around the extremes of the coordinates.

Table II compares the NN inferences with the ground truth using several metrics: trajectory correlation, and RMSs of speed and acceleration. The speed and acceleration RMS are calculated as the square root of the mean square of the first and the second derivatives of the inferred locations, respectively. The correlations of both the TCN and CAPSNET are slightly lower than that of the 1D-CNN (87.5 %), but while the TCN RMSs of speed and acceleration agree very well with the ground truth, the CAPSNET inference appears to be too smooth.

The upper half of Table III compares two metrics of the inference accuracy of the TCN optimized by NAS, the "TCN (NAS)" in Table I, used as baseline, with its accuracy after refining its training with several KD techniques (without changing the NN

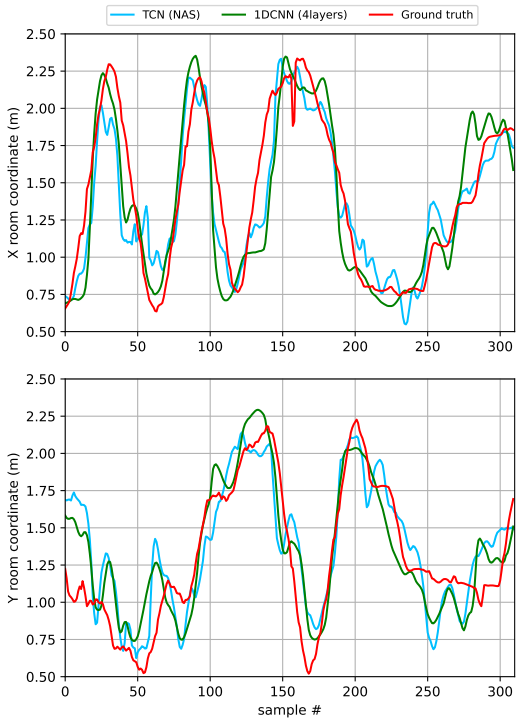

Fig. 4. Ground truth and inference of trajectory coordinates X (top) and Y (bottom) by the best neural network in [10], 1D convolutional neural network (1D-CNN) (4 layers), and our temporal convolutional network (TCN) optimized using neural architecture search (NAS).

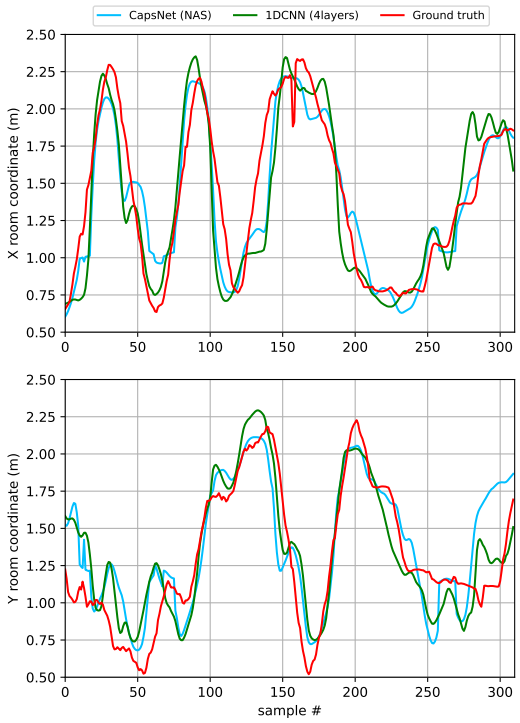

Fig. 5. Ground truth and inference of trajectory coordinates X (top) and Y (bottom) by the best neural network in [10], 1D convolutional neural network (1D-CNN) (4 layers), and our capsule network (CAPSNET) optimized using neural architecture search (NAS).

TABLE II

MOVEMENT INFERENCE QUALITY OF OUR TCN AND CAPSNET COMPARED TO [10] (HIGHLIGHTED) AS GROUND TRUTH CORRELATION, AND ROOT MEAN SQUARE OF SPEED AND ACCELERATION

Model	Correlation (%)	RMS speed (m/s)	RMS acc. ($\mathrm{m/s^2}$)
Ground truth	100.0	0.180	0.333
1D-CNN (2 layers)	83.3	0.157	0.172
1D-CNN (4 layers)	**87.5**	0.162	0.187
1D-CNN (6 layers)	84.5	**0.176**	0.259
LSTM (bidirectional)	84.0	0.133	0.129
TCN (NAS)	86.0	**0.180**	**0.347**
CAPSNET (NAS)	**87.1**	0.164	0.384

TABLE III

GROUND TRUTH CORRELATION, MEAN SQUARED ERROR (MSE) AND PERCENTAGE IMPROVEMENTS, AND AVERAGE EUCLIDEAN DISTANCE ERROR (ADE) OF TEMPORAL CONVOLUTIONAL NETWORK (TCN) AND CAPSULE NETWORK (CAPSNET) TRAINED FROM SCRATCH (BASELINES), AND THEN REFINED BY SEVERAL KNOWLEDGE DISTILLATION (KD) TECHNIQUES

Optimization	Correlation (%)	MSE ($\mathrm{m^2}$)	MSE gain (%)	ADE (m)
TCN (NAS) baseline (2034 parameters)	86.0	0.065	N/A	0.309
TCN with KD (2034 parameters)				
• Imitation loss regularization	88.0	0.061	6.0	0.301
• Attentive imitation loss	88.4	**0.055**	**15.4**	**0.285**
• Teacher loss as upper bound	**88.5**	0.056	13.8	0.286
CAPSNET (NAS) baseline (5996 parameters)	87.1	0.063	N/A	0.303
CAPSNET with KD (5996 parameters)				
• Imitation loss regularization	87.0	0.061	3.2	0.304
• Attentive imitation loss	87.6	0.061	3.2	0.302
• Teacher loss as upper bound	**87.9**	**0.059**	**6.0**	**0.298**

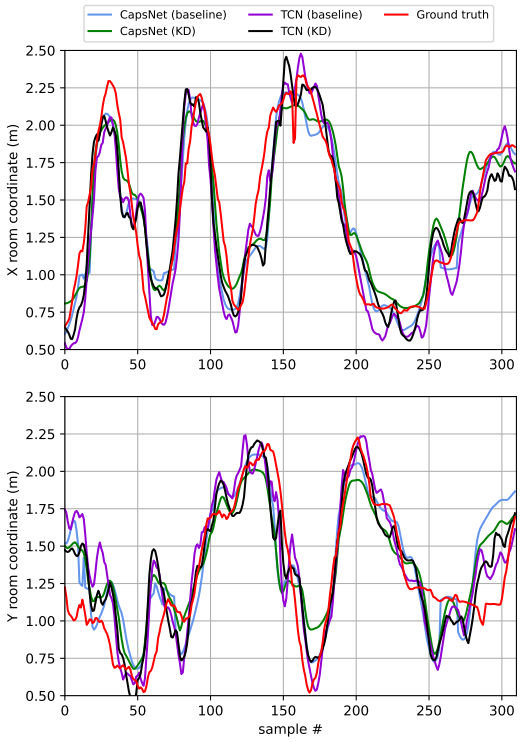

Fig. 6. Ground truth and inference of trajectory coordinates X (top) and Y (bottom) by both capsule network (CAPSNET) and temporal convolutional network (TCN), first trained from scratch (baseline), then optimized by knowledge distillation (KD) technique with the best performance.

TABLE IV

COMPARISON OF OUR BEST PERFORMING TEMPORAL CONVOLUTIONAL NETWORK (TCN) AND CAPSULE NETWORK (CAPSNET) WITH THE BEST PERFORMING 1D CONVOLUTIONAL NEURAL NETWORK (1D-CNN) IN [10]

Model	Number of parameters	Parameter fraction (%)	MSE ($\mathrm{m^2}$)	MSE gain (%)
1D-CNN (4 layers)	7618	N/A	0.063	N/A
TCN	**2034**	**26.7**	**0.055**	**12.7**
CAPSNET	5996	78.7	0.059	6.3

structure): imitation loss regularization, attentive imitation loss, and teacher loss as upper bound. The NN used as teacher in KD is another TCN optimized for high accuracy ($\mathrm{MSE} = 0.059\,\mathrm{m^2}$ and $\mathrm{ADE} = 0.29\,\mathrm{m}$), but with higher resource consumption (7242 parameters). All KD refinements improve all inference metrics compared to the baseline TCN, with the attentive imitation loss KD strategy achieving the best results (15.4 % better MSE and 2.9 % better movement correlation).

The bottom half of Table III reports analogous comparisons for CAPSNET using as teacher a large CAPSNET model with 6734 parameters, which achieved $\mathrm{MSE} = 0.061\,\mathrm{m^2}$ and $\mathrm{ADE} = 0.299\,\mathrm{m}$. The student CAPSNET is the "CAPSNET (NAS)" in Table I. The teacher loss as upper bound KD strategy gives the best results, reducing the inference MSE to $0.059\,\mathrm{m^2}$, or 6.0 % better than the baseline.

Fig. 6 shows the plots of the ground truth coordinates X and Y compared to the inferences of the TCN and CAPSNET models optimized by NAS only (baselines), and then the best performing models optimized by KD. We notice that the inference of the CAPSNET optimized by both NAS and KD tends to cover better the extremes of the coordinates, avoiding oscillations (especially in the last part of the Y coordinate). The TCN optimized by both NAS and KD, which has the best inference MSE, tends to reduce the amplitude of the oscillations of the TCN model optimized only by NAS, especially in the Y coordinate and in the last part of the X coordinate.

Table IV summarizes the performance improvements of the optimization of the TCN and CAPSNET compared to the best performing 1D-CNN in [10]. The TCN appears to be the most suited architecture for indoor person localization and tracking using capacitive sensors, obtaining good noise rejection and infer the trajectory better, by 12.7 %, than the reference 1D-CNN in [10] using just a small fraction, 26.7 %, of the resources.

VI. CONCLUSIONS

1D convolutional neural networks (1D-CNNs) have been shown to excel in processing experimental time series data for

indoor human localization and tracking from noisy long-range single-plate capacitive sensors [10]. However, neural networks that do not use the convolutional neural network (CNN) max pooling operations can exceed the accuracy of 1D-CNNs with fewer resources, especially when they are optimized with a combination of state-of-the-art techniques.

Temporal convolutional networks (TCNs) and capsule networks (CAPSNETs) are designed to process time series data without using the CNN max pooling operations. Combining hyperparameter optimization with neural architecture search (NAS) and training optimization with knowledge distillation (KD) can significantly improve both their accuracy and resource consumption. With NAS optimization, the CAPSNET was as accurate as the 1D-CNN, but used only 78.7 % of the resources. Then the KD optimization improved its accuracy by 6.3 %. Similarly, the TCN was almost as accurate as the 1D-CNN after the NAS optimization, but used only a small fraction, 26.7 %, of the resources, while the KD optimization improved its accuracy by 12.7 %.

TCNs seem better suited for indoor human localization using noisy capacitive sensors than CAPSNETs and 1D-CNNs, benefiting most from combined NAS and KD optimizations to achieve the best accuracy with much less resource consumption.

REFERENCES

[1] D. Xu, Y. Wang, B. Xiong, and T. Li, "MEMS-based thermoelectric infrared sensors: A review," *Frontiers of Mechanical Engineering*, vol. 12, no. 4, pp. 557–566, 2017.

[2] C.-H. Chu, C.-H. Wang, C.-K. Liang, W. Ouyang, *et al.*, "High-accuracy indoor personnel tracking system with a ZigBee wireless sensor network," in *2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks*, IEEE, 2011, pp. 398–402.

[3] J. Fortes and D. Kocur, "Solutions of mutual shadowing effect between people tracked by UWB radar," in *2013 IEEE International Conference on Microwaves, Communications, Antennas and Electronic Systems (COMCAS 2013)*, IEEE, 2013, pp. 1–5.

[4] M. Kuki, H. Nakajima, N. Tsuchiya, and Y. Hata, "Multi-human locating in real environment by thermal sensor," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, 2013, pp. 4623–4628.

[5] J. Bordoy, J. Wendeberg, C. Schindelhauer, and L. M. Reindl, "Single transceiver device-free indoor localization using ultrasound body reflections and walls," in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, 2015, pp. 1–7.

[6] Y. Kim, S. Ha, and J. Kwon, "Human detection using Doppler radar based on physical characteristics of targets," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 2, pp. 289–293, 2014.

[7] J. Smith, T. White, C. Dodge, J. Paradiso, *et al.*, "Electric field sensing for graphical interfaces," *IEEE Computer Graphics and Applications*, vol. 18, no. 3, pp. 54–60, 1998.

[8] A. Arshad, S. Khan, A. Z. Alam, R. Tasnim, *et al.*, "An activity monitoring system for senior citizens living independently using capacitive sensing technique," in *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, IEEE, 2016, pp. 1–6.

[9] G. Subbicini, L. Lavagno, and M. T. Lazarescu, "Drift Rejection Differential Frontend for Single Plate Capacitive Sensors," *IEEE Sensors Journal*, vol. 22, no. 16, pp. 16 141–16 149, 2022.

[10] O. B. Tariq, M. T. Lazarescu, and L. Lavagno, "Neural Networks for Indoor Human Activity Reconstructions," *IEEE Sensors Journal*, vol. 20, no. 22, pp. 13 571–13 584, 2020.

[11] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, *et al.*, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, pp. 1–74, 2021.

[12] S. Bai, J. Z. Kolter, and V. Koltun, *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*, 2018.

[13] K. Guirguis, C. Schorn, A. Guntoro, S. Abdulatif, *et al.*, "SELD-TCN: Sound event localization & detection via temporal convolutional networks," in *2020 28th European Signal Processing Conference (EUSIPCO)*, IEEE, Jan. 2021.

[14] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, *et al.*, *Temporal Convolutional Networks for Action Segmentation and Detection*, 2016.

[15] Y. Kim, P. Wang, Y. Zhu, and L. Mihaylova, "A Capsule Network for Traffic Speed Prediction in Complex Road Networks," in *2018 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, 2018, pp. 1–6.

[16] K. Suri and R. Gupta, "Continuous sign language recognition from wearable IMUs using deep capsule networks and game theory," *Computers & Electrical Engineering*, vol. 78, pp. 493–503, 2019.

[17] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[18] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[19] M. R. U. Saputra, P. P. B. de Gusmao, Y. Almalioglu, A. Markham, *et al.*, *Distilling Knowledge From a Deep Pose Regressor Network*, 2019.

[20] D. Savio and T. Ludwig, "Smart Carpet: A Footstep Tracking Interface," *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, vol. 2, pp. 754–760, 2007.

[21] C. Lauterbach, A. Steinhage, and A. Techmer, "Large-area wireless sensor system based on smart textiles," in *International Multi-Conference on Systems, Signals & Devices*, 2012, pp. 1–2.

[22] M. Valtonen, J. Maentausta, and J. Vanhala, "TileTrack: Capacitive human tracking using floor tiles," in *2009 IEEE International Conference on Pervasive Computing and Communications*, 2009, pp. 1–10.

[23] W. Buller and B. Wilson, "Measuring the Capacitance of Electrical Wiring and Humans for Proximity Sensing with Existing Electrical Infrastructure," in *2006 IEEE International Conference on Electro/Information Technology*, 2006, pp. 93–96.

[24] P. Srinivasan, D. Birchfield, G. Qian, and A. Kidané, "A Pressure Sensing Floor for Interactive Media Applications," in *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ser. ACE '05, Valencia, Spain: Association for Computing Machinery, 2005, pp. 278–281.

[25] V. Bellavista-Parent, J. Torres-Sospedra, and A. Pérez-Navarro, "Comprehensive Analysis of Applied Machine Learning in Indoor Positioning Based on Wi-Fi: An Extended Systematic Review," *Sensors*, vol. 22, no. 12, 2022.

[26] Q. Ye, X. Fan, G. Fang, H. Bie, *et al.*, "CapsLoc: A Robust Indoor Localization System with WiFi Fingerprinting Using Capsule Networks," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[27] B. Jia, J. Liu, T. Feng, B. Huang, *et al.*, "TTSL: An indoor localization method based on Temporal Convolutional Network using time-series RSSI," *Computer Communications*, vol. 193, pp. 293–301, 2022.

[28] T. Grosse-Puppendahl, C. Holz, G. Cohn, R. Wimmer, *et al.*, "Finding Common Ground: A Survey of Capacitive Sensing in Human-Computer Interaction," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI '17, Denver, Colorado, USA: Association for Computing Machinery, 2017, pp. 3293–3315.

[29] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, *et al.*, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15, Montreal, Canada: MIT Press, 2015, pp. 802–810.

[30] J. Bradbury, S. Merity, C. Xiong, and R. Socher, *Quasi-Recurrent Neural Networks*, 2016.

[31] S. Chang, Y. Zhang, W. Han, M. Yu, *et al.*, *Dilated Recurrent Neural Networks*, 2017.

[32] El Alaoui-Elfels, Omaima and Gadi, Taoufiq, "From Auto-encoders to Capsule Networks: A Survey," *E3S Web Conf.*, vol. 229, p. 01 048, 2021.

[33] H. Jin, F. Chollet, Q. Song, and X. Hu, "AutoKeras: An AutoML Library for Deep Learning," *Journal of Machine Learning Research*, vol. 24, no. 6, pp. 1–6, 2023.

[34] F. Chollet *et al.*, *Keras*, https://keras.io, 2015.

[35] I. Paik, T. Kwak, and I. Kim, "Capsule networks need an improved routing algorithm," in *Asian Conference on Machine Learning*, PMLR, 2019, pp. 489–502.

[36] S. R. Venkataraman, S. Balasubramanian, and R. R. Sarma, "Iterative collaborative routing among equivariant capsules for transformation-robust capsule networks," *arXiv preprint arXiv:2210.11095*, 2022.

[37] Z. Zhao and S. Cheng, "Capsule networks with non-iterative cluster routing," *Neural Networks*, vol. 143, pp. 690–697, 2021.