Deep Convolutional Neural Networks for Massive MIMO Fingerprint-Based Positioning

Joao Vieira, Erik Leitinger, Muris Sarajlic, Xuhong Li, Fredrik Tufvesson
Dept. of Electrical and Information Technology, Lund University, Sweden
joao.vieira@eit.lth.se

***Abstract*—This paper provides an initial investigation on the application of convolutional neural networks (CNNs) for fingerprint-based positioning using measured massive MIMO channels. When represented in appropriate domains, measured massive MIMO channels have a sparse structure which can be efficiently learned by CNNs for positioning purposes. We evaluate the positioning accuracy of state-of-the-art CNNs with channel fingerprints generated from a channel model with a rich clustered structure: the COST 2100 channel model. We find that moderately deep CNNs can achieve fractional-wavelength positioning accuracies, provided that an enough representative data set is available for training.**

I. Introduction

In its originally conceived form, massive MIMO uses a large number of base station (BS) antennas together with *measured* channel state information (CSI) to multiplex users terminals spatially [1]. Measured CSI is essential to yield spectrally efficient communications, but it can also be a key enabler to achieve highly-accurate terminal positioning, where down to centimeter order accuracy may be required in some 5G applications, e.g., autonomous driving [2]. Explained briefly, since positioning is a spatial inference problem, it makes sense to use large antenna arrays that oversample the spatial dimension of a wireless channel - thus benefiting from, e.g., increased angular resolution, resilience to small-scale fading, and array gain effects - to aid the positioning task.

Our aim in this work is to perform fingerprint-based positioning based on measured massive MIMO channels. More especifically, we are interested to learn

$$f^{-1} : \{s\left(\mathbf{Y}_i\right)\} \rightarrow \{\mathbf{x}_i\}, \tag{1}$$

that is, the inverse of the underlying function $f(\cdot)$ that maps each label $\mathbf{x}_i$ to its respective observation $s\left(\mathbf{Y}_i\right)$, from a training set $\{s\left(\mathbf{Y}_i\right), \mathbf{x}_i\}_{i=1}^{N_{\text{Train}}}$. Here the label $\mathbf{x}_i \in \mathbb{R}^{2\text{x}1}$ is the 2-dimensional terminal coordinate of training observation i, and $s\left(\mathbf{Y}_i\right) \in \mathbb{C}^{D_1\text{x}\ldots\text{x}D_D}$ is its associated *measured*, but transformed, channel fingerprint. We note that the main point of the transformation $s(\cdot)$ is to obtain a sparse representation for $s\left(\mathbf{Y}_i\right)$. This is motivated in detail in Sec. II-B. For now, we remark that the sparse transformations considered in this work are bijective, and thus yield no information loss.

Our proposal to approximate (1) is by means of deep CNNs. Deep neural networks provide state-of-the-art learning machines that yield the most learning capacity from all machine learning approaches [3], and lately have been very successful in image classification tasks. Just like most relevant information for an image classification task is sparsely distributed at some locations of the image [3], measured channel snapshots $\mathbf{Y}_i$ have, when represented in appropriate domains, a sparse structure which - from a learning perspective - resemble that of images. This sparse channel structure can be learned by CNNs and therefore used for positioning purposes. To the best of the authors' knowledge there is no prior work on this matter.

II. Channel Fingerprinting and Pre-Processing

A. Channel Fingerprinting

In this work, we assume a BS equipped with a linear M-antenna array made of omnidirectional $\lambda/2$-spaced elements, and that narrowband channels sampled at N_F equidistant frequency points are used for positioning. With that, the dimensions of each channel fingerprint $\mathbf{Y}_i$ (which, as it will be seen later, are equal to those of the transformed fingerprint $s(\mathbf{Y}_i) \in \mathbb{C}^{D_1\text{x}\ldots\text{x}D_D}$) are

$$D_1 = M,\ D_2 = N_F,\ \text{and}\ D_d = 1 \text{ with } 3 \leq d \leq D.$$

Given a terminal position, its associated fingerprint is generated through $f(\cdot)$, i.e. the inverse of the function we wish to learn. We implement $f(\cdot)$ using the COST 2100 channel model - the structure of which is illustrated by Fig. 1 - under the parametrization proposed in [4]. This parametrization is further detailed in Sec. IV. It is important to note that, in this work, $f(\cdot)$ is implemented as a bijective deterministic map, i.e., there is only one unique fingerprint per position.

B. Motivation for CNNs and Sparse Input Structures

CNNs are efficient learning machines given that their inputs meet the following two structural assumptions:

1) most relevant information features are sparsely distributed in the input space;
2) the shape of most relevant information features is invariant to their location in the input space, and are well captured by a finite number of kernels.

From a wireless channel point-of-view, these assumptions apply well when channels snapshots (i.e., the CNN inputs) are represented in domains that yield a sparse structure [5]. For example, in the current case study, sparsity is achieved by representing $\mathbf{Y}_i$ in its, so-called, angular-delay domains, see Fig. 1. Trivially, $s(\cdot)$ can take the form of a two-dimensional discrete Fourier transform, i.e.,

$$s(\mathbf{Y}_i) = \mathbf{F}\,\mathbf{Y}_i\,\mathbf{F}^H. \tag{2}$$

If specular components of the channel, which are typically modeled by Dirac delta functions [5], are seen as the information basis for positioning, then the two structural assumptions

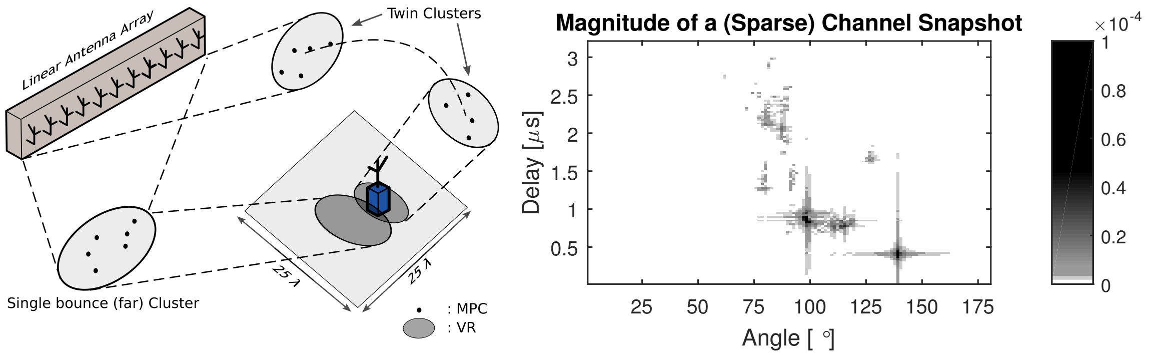

Fig. 1. Left - Link setup considered in this work: an M-linear BS array positioning one single-antenna terminal in a confined square area. Channel realizations are generated through the COST 2100 MIMO channel model. This geometry-based stochastic channel model is composed by different types of clusters of multipath components (MPCs) that illuminate certain visibility regions (VRs) of an area. Right - Example of the magnitude of a channel snapshot represented in a sparse domain. Such channel channel has a rich structure that can be learned by a CNN for positioning purposes.

of the CNNs inputs listed above are met. The same applies, if instead, clusters of multipath components are seen as the information features for positioning.

To finalize, we remark that the current case study can be extended to more generic/higher-dimensionality fingerprints, e.g., when $D_d > 1 \, \forall \, d \geq 3$. In any case, the key is the ability to obtain a sparse representation for $s\left(\mathbf{Y}_i\right) \in \mathbb{C}^{D_1 \text{x} \dots \text{x} D_D}$.

III. Deep CNN Architecture

After the input layer, which takes the transformed snapshots $s(\mathbf{Y}_i)$, the structure of CNNs we use employs a cascade of L convolutional-activation-pooling (CAP) layers. Each CAP layer is composed by: *i)* a convolutional operation of its input with K convolutional Kernels, *ii)* a (pre-defined) non-linear transformation, i.e., activation function, and *iii)* a pooling layer, respectively [3]. Then, a fully-connected layer, following the L CAP layers, finalizes the CNN by producing the position estimate of $\mathbf{x}$, namely, $\mathbf{t} \in \mathbb{R}^{2\text{x}1}$.

The CNN learns its parameters (i.e. weights and biases), which we stack in $\boldsymbol{\theta}$ for latter use, in order to make $\mathbf{t}$ the best approximation of $\mathbf{x}$. Since we address positioning as a regression problem, we use the squared residuals averaged over the training set as the optimizing metric. Hence, the parameters estimates are given by

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \;\; \frac{\beta}{2}\boldsymbol{\theta}^T\boldsymbol{\theta} + \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} (\mathbf{x}_i - \mathbf{t}_i(\boldsymbol{\theta}))^2. \tag{3}$$

A Tikhonov penalty term is added to harvest the benefits of regularization in CNNs - β is its associate hyper-parameter.

IV. Numerical Results

A. Simulation Setup

The spatial setup used in our experiments is illustrated by Fig. 1: the terminal is constrained to be in a square area $\mathcal{A}$ of 25×25 wavelengths. Channel fingerprints are obtained in this area through the COST 2100 channel model under the 300 MHz parameterization (e.g., for path-loss and cluster-based parameters) established in [4]. The remaining parameters are shown in Table I, and the other CNNs hyper-parameters, i.e. L and K, are varied during the simulations.

TABLE I
Channel and CNNs parameters

Parameter	**Variable**	**Value**
Carrier frequency	f_c	300 MHz
Bandwidth	W	20 MHz
# Frequency points	N_F	128
# BS antennas	M	128
First BS antenna coordinate	$\mathbf{B}_1$	$[-200\lambda \;\; -200\lambda]^T$
Last BS antenna coordinate	$\mathbf{B}_M$	$[-200\lambda \;\; -200\lambda + \frac{(M-1)\lambda}{2}]^T$
Tikhonov hyper-parameter	β	10^{-3}
Kernel angular length [°]	S_1	9.8
Kernel delay length [μs]	S_2	0.175
Pooling windows length	N_1 and N_2	2 ”bins”

The closest and furthest coordinate points of $\mathcal{A}$ with respect to the first BS antenna are:

$$\mathbf{u}_c = [-12.5\lambda \;\; -12.5\lambda]^T \text{ and } \mathbf{u}_f = [12.5\lambda \; 12.5\lambda]^T,$$

respectively (i.e., the user is at least $||\mathbf{u}_c - \mathbf{B}_1||/\lambda$ wavelengths away from the first BS antenna). The coordinates of these two spatial points implicitly define the relative orientation of the linear array with the area $\mathcal{A}$. Similarly, the upcoming performance analysis is done by means of the normalized root mean-squared error (NRMSE), where the mean consists of the average over the test sets samples, respectively. Thus, we have

$$\text{NRMSE} = \frac{1}{\lambda}\sqrt{\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} (\mathbf{x}_i - \mathbf{t}_i(\boldsymbol{\theta}))^2}.$$

This error metric has an understandable physical intuition as it shows how the error distance relates to the wavelength.

The CNN training and testing is described as follow:

1) First, the training set is obtained by fingerprinting a 2-dimensional uniformly-spaced (thus, deterministic) grid of positions spanning the totality of $\mathcal{A}$. The impact of the sampling density is discussed in Sec. IV-C.
2) For the test set, the position at which fingerprints are obtained is modeled as a random variable drawn from a uniform distribution with support $\mathcal{A}$.

Note that, if the CNN cannot use the available fingerprints for training, then the position estimator is $\mathbb{E}\{\mathbf{x}\} = \mathbf{0}$. Its

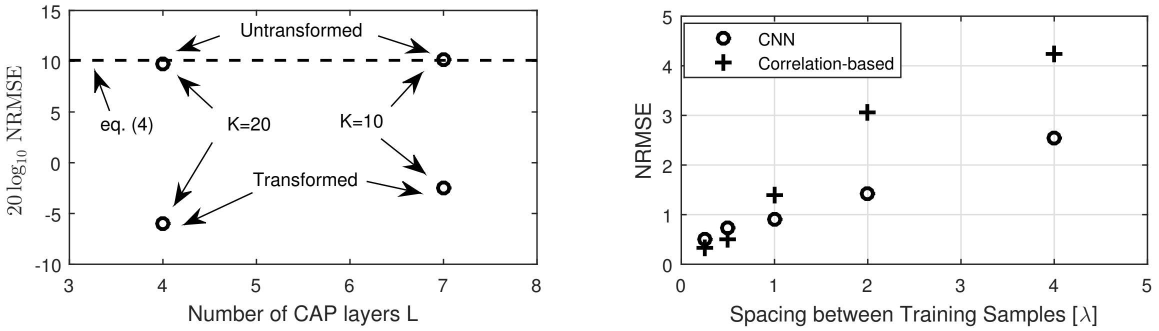

Fig. 2. Left - NRMSE obtained by CNNs under different parameterizations. The upper horizontal line corresponds of the reference level (4). Here we only report the test error, since a similar error value was obtained during training (i.e., no overfitting exists). Right - NRMSE obtained by different positioning approaches for different spacings between samples of the uniform training grid.

NRMSE, for the current case study, is given by

$$\mathrm{NRMSE}^{\mathrm{ref}}(\mathcal{A}) = \frac{1}{\lambda}\sqrt{\frac{1}{\int_{\mathcal{A}} \partial \mathbf{d}} \int_{\mathcal{A}} \left(\mathbf{d} - \mathbb{E}\{\mathbf{x}\}\right)^2 \partial \mathbf{d}} \approx 10.2, \tag{4}$$

which use as a reference level in the analysis.

To finalize, we also contrast our CNN results against a standard non-parametric fingerprinting approach [6], for benchmarking purposes. Seeing a training fingerprint as a function of its position, i.e. $\mathbf{Y}_i(\mathbf{x}_i)$, this approach computes the position from a new fingerprint $\mathbf{Y}_{\mathrm{new}}$ through a grid-search over normalized correlations as

$$\hat{\mathbf{x}}_i = \underset{\mathbf{x}_i \in \{\mathbf{x}_i\}_{i=1}^{N_{\mathrm{train}}}}{\mathrm{argmax}} \quad \frac{|\operatorname{Tr}\{\mathbf{Y}_i(\mathbf{x}_i)^H \mathbf{Y}_{\mathrm{new}}\}|}{\sqrt{|\operatorname{Tr}\{\mathbf{Y}_i(\mathbf{x}_i)^H \mathbf{Y}_i(\mathbf{x}_i)\} \operatorname{Tr}\{\mathbf{Y}_{\mathrm{new}}^H \mathbf{Y}_{\mathrm{new}}\}|}}. \tag{5}$$

Compared to the use of CNNs, a main drawback of this approach is its computational complexity order, $\mathcal{O}(MN_F^2 N_{\mathrm{train}})$, which depends on the size of the training set;

B. Proof-of-Concept and Accuracy for Different CNN Parametrizations

Here, we report the positioning results when the spacing between neighbor training fingerprints is $\lambda/4$. Fig. 2 (left) illustrates the positioning accuracy for different cases of CNN parameterizations. First, and as a sanity check, we see that for the same parameterizations, a network fed with untransformed inputs (i.e., $s\left(\mathbf{Y}_i\right) = \mathbf{Y}_i$) cannot effectively learn the channel structure for positioning purposes - the order of magnitude of the positioning error is similar to (4). However, with transformed inputs, fractional-wavelength positioning can be achieved in both network settings, with the lowest achieved test NRMSE being of about $-6\mathrm{dB} \approx 1/2$ of a wavelength. This showcases the capabilities of CNNs to learn the structure of the channel for positioning purposes. We remind that such positioning accuracies are attained with only 20 MHz of signaling bandwidth, which suggests that CNNs can efficiently trade-off signal bandwidth by BS antennas and still achieve very good practical performance. Decreasing the error further than fractional-wavelength ranges becomes increasingly harder due to the increased similarities of nearby fingerprints - such range approaches the coherence distance/bandwidth of the channel.

C. Accuracy for Different Training Grids

To finalize, we analyze the impact of spatial sampling during training. For benchmarking, we contrast the CNN performance with the performance of the correlation-based classifier (5). We use the CNN model that attained the lowest MSE in Fig. 2 (left), namely, the model with $L = 4$ and $K = 20$. Fig. 2 (right) contrasts the NRMSEs obtained from a CNN and the correlation-based classifier (5), against spatial sampling in the training set. Overall, both approaches are able to attain fractional-wavelength accuracies at smaller training densities. Noticeably, the CNN tend to behave better than (5) for less dense training sampling. Given that (5) does not have interpolation abilities, this result is closely connected with the inherent interpolation abilities of the CNNs. The fact that the CNNs achieve similar, or even superior performance compared to standard non-parametric approaches while having attractive implementation complexity further corroborates their use in fingerprint-based localization systems.

V. Takeaways and Further Work

We have investigated a novel approach for massive MIMO fingerprint-based positioning by means of CNNs and measured channel snapshots. CNNs have a feedforward structure that is able to compactly summarize relevant positioning information in large channel data sets. The positioning capabilities of CNNs tend to generalize well, e.g. in highly-clustered propagation scenarios with or without LOS, thanks to their inherent feature learning abilities. Proper design allows fractional-wavelength positioning to be obtained under real-time requirements, and with low signal bandwidths.

The current investigation showcased some of the potentials of CNNs for positioning using channels with a complex structure. However, the design of CNNs in this contexts should be a matter of further investigation, in order to be able to deal with real-world impairments during the fingerprinting process. In this vein, some questions raised during this study are, for example, *i)* how to achieve a robust CNN design that is able to deal with impairments such as measurement and labeling noise, or channel variations that are not represented in the training set, or *ii)* how to design complex-valued CNNs that perform well and are robust during optimization.

References

[1] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, “Massive MIMO for next generation wireless systems,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, 2014.

[2] Z. Chaloupka, “Technology and Standardization Gaps for High Accuracy Positioning in 5G,” *IEEE Communications Standards Magazine*, vol. 1, no. 1, pp. 59–65, March 2017.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[4] M. Zhu, G. Eriksson, and F. Tufvesson, “The COST 2100 Channel Model: Parameterization and Validation Based on Outdoor MIMO Measurements at 300 MHz,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 888–897, February 2013.

[5] A. Molisch, *Wireless Communications*, ser. Wiley - IEEE. Wiley, 2010.

[6] Z. H. Wu, Y. Han, Y. Chen, and K. J. R. Liu, “A time-reversal paradigm for indoor positioning system,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1331–1339, April 2015.