electronics

Article

Low-Complexity Convolutional Neural Network for Channel Estimation

Simona Sibio [1,*], Cristian Sestito [2], Souheil Ben Smida [1], Yuan Ding [1] and George Goussetis [1]

[1] Institute of Sensors Signals and Systems, Heriot-Watt University, Edinburgh EH14 4AS, UK
[2] Centre for Electronics Frontiers, Institute for Integrated Micro and Nano Systems, School of Engineering, University of Edinburgh, Edinburgh EH8 9YL, UK; csestito@ed.ac.uk
* Correspondence: ss870@hw.ac.uk

Abstract: This paper presents a deep learning algorithm for channel estimation in 5G New Radio (NR). The classical approach that uses neural networks for channel estimation requires more than one stage to obtain the full channel matrix. First, the channel has to be constructed by the received reference signal, and then, the precision is improved. In contrast, to reduce the computational cost, the proposed neural network method generates the channel matrix from the information captured from a few subcarriers along the slot. This information is extrapolated by applying the Least Square technique only on the Demodulation Reference Signal (DMRS). The received DMRS placed in the grid can be seen as a 2D low-resolution image and it is processed to generate the full channel matrix. To reduce complexity in the hardware implementation, the convolutional neural network (CNN) structure is selected. This solution is analyzed comparing the Mean Square Error (MSE) and the computational cost with other deep learning-based channel estimation, as well as the traditional channel estimation methods. It is demonstrated that the proposed neural network delivers substantial complexity savings and favorable error performance. It reduces the computational cost by an order of magnitude, and it has a maximum error discrepancy of 0.018 at 5 dB compared to Minimum Mean Square Error (MMSE) channel estimation.

Keywords: channel estimation; computational cost; convolutional neural network; MIMO system; MSE analysis

Citation: Sibio, S.; Sestito, C.; Smida, S.B.; Ding, Y.; Goussetis, G. Low-Complexity Convolutional Neural Network for Channel Estimation. *Electronics* **2024**, *13*, 4537. https://doi.org/10.3390/electronics13224537

Academic Editor: Dimitra I. Kaklamani

Received: 30 September 2024
Revised: 11 November 2024
Accepted: 15 November 2024
Published: 19 November 2024

1. Introduction

Due to the high variability of the wireless channel, its estimation is recognized as one of the most challenging techniques in many cellular use cases, such as Reconfigurable Intelligent Surface (RIS), mmWaves, massive Multiple-Input Multiple-Output (MIMO), and Open Radio Access Network (O-RAN) for the Radio Unit. To support more stringent requirements of next generation systems, the traditional channel estimation techniques must be improved, as they become a bottleneck due to high complexity or limited accuracy. Conventional methods, such as Least Square (LS) and Minimum Mean Square Error (MMSE), follow mathematical models and are blind to the underlying characteristics of the channels [1]. Commonly, LS offers the lowest accuracy in many scenarios, but has the benefit of not requiring prior information. Thus, it is still considered a valuable choice when implementation is considered [2]. Alternatively, MMSE minimizes the estimation error, but has a higher computational cost requiring channel statistical information [3].

By virtue of the ability to adapt quickly in various environments, Deep Learning (DL) techniques have gained interest in many aspects of wireless communication systems showing promising results [4]. Since DL is purely data-driven, the networks are optimized over large training datasets and can then provide very high levels of accuracy [5]. It has been demonstrated that neural network-based channel estimation outperforms the MMSE technique, as shown in [3,6]. In particular, ref. [6] demonstrates error improvement up to

2 orders of magnitude compared to MMSE channel estimation, while [3] highlights the efficacy with different Doppler frequencies in the transmission channel model. However, a major concern regarding DL solutions is their effectiveness in real-world scenarios, and [7] demonstrates the validity of orthogonal approximate message passing (OAMP) using DL methods and how this approach can improve performance.

Generally, in the literature, the neural network design relies on one of three types of Deep Neural Network (DNN) models, namely, Fully Connected (FCDNNs), Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). Despite the fact that the performance of the neural network can vary with the design, FCDNNs are usually applied for classification problems, and are less suitable for channel estimation compared to the CNNs and RNNs, as demonstrated in [8]. RNNs are able to keep track of the information history, which is particularly suitable for channel estimation because the information on one symbol is propagated along the slot, as shown in [9]. In particular, the Long Short-Term Memory (LSTM), one of the most common RNNs, has shown the best results in terms of bit error rate (BER) and mean square error (MSE). Whilst the LSTM-based architecture reaches the lowest error rate, it unfortunately requires more complex operations. Therefore, this architecture is not considered a good fit for hardware technology, such as System on Chip (SoC) or Versal [10]. CNNs are traditionally used for image denoising. They are therefore favorably applied to the noisy channel information in order to obtain a clearer version as proved with ChannelNet, one of the first neural networks to be used for channel estimation [11]. Two neural networks, specifically image super-resolution (SR) and image restoration (IR), are used to enhance channel estimation, outperforming the MMSE technique. Another approach is investigated in [12]; the overall proposal includes two neural networks, but one in the user equipment (UE) and the other is in the base station (BS). This setup enables the UE to gather Channel State Information, compressing it to minimize transmission overhead, which results in improved MSE performance at the BS. Similar in [13,14], it exploits the attention mechanism to focus on the critical features to yield the best channel estimate. Indeed, in [13], first, multiple FCDNN-layers are used for frequency-aware pilot design, and then a hybrid neural network is placed in the UE to acquire the channel knowledge. The latter is composed of a FCDNN-layer per subcarrier, three CNN-layers, and a Non-Local Attention module. Instead [14], proposes an encoder (FCDNN, normalization layers, and an attention algorithm) and a convolutional architecture decoder composed of seven layers.

Despite the advantages of an accurate estimated channel using neural networks, all of the above cited works require different stages to acquire the full channel matrix (e.g., two neural networks or one neural network and an interpolation along subcarriers). A different solution is considered in [15] using one CNN, called fast super-resolution (FSRCNN). The latter uses eight CNN-layers to produce the channel matrix from the pilot information. However, the pilot information is dimensioned though zero-padding in the input, and each layer uses multiple channels. These two factors lead to an unnecessary growth of the complexity and possible overfitting.

All of the above considerations result in a critical computational cost when implemented in hardware.

To address this concern, we propose a low-complexity CNN which treats the channel information in the Demodulation Reference Signal (DMRS) as a low-resolution noisy 2D image. The extrapolation of the full channel matrix is performed by gradually adjusting the padding at each layer. Because the channel is described by complex values, this aspect is handled by applying the neural network in the real and imaginary parts, separately. The contributions of this paper are summarized as follows:

1. The CNN is used for increasing the resolution of the 2D channel image. Only one CNN is implemented to generate the full channel grid from the DMRS.
2. The computational cost is significantly reduced compared to other channel estimation techniques.

3. The improvements of the proposed approach are demonstrated by virtue of analyzing several DL-channel estimation techniques in terms of complexity and MSE in a 5G NR system. The stage-by-stage complexity analysis offers a tool for assessing the complexity of any neural network incorporating LSTM, FCDNN, or CNN layers.

This paper is organized as follows. In Section 2, the 5G NR system model is described, and the challenges of channel estimation are discussed. In Section 3, the traditional channel estimations are introduced while highlighting advantages and drawbacks. In Section 4, the focus is on the DL-channel estimations with particular attention to the proposed low complexity CNN. In order to evaluate all of the different channel estimation methods, the computational costs are compared in Section 5, and MSE analysis in a 5G NR simulation is presented in Section 6. Finally, the conclusions are drawn in Section 7.

2. 5G NR System Model

Considering the downlink in a single-cell environment, the waveforms are sent by the transmitter equipped with M antennas over the channel, and a corrupted version of them are collected by K antennas at the receiver. The propagation over the channel can bring noise, attenuation, distortion, fading, and interference [16]. Mathematically, this is Equation (1).

$$Y = \boldsymbol{H}\, X + N \tag{1}$$

where $Y = (y_1\ y_2\ \ldots\ y_K)^T \in \mathbb{C}^{\boldsymbol{K}\times\boldsymbol{1}}$, is the received signal in vector form. The channel environment is captured by $\boldsymbol{H} \in \mathbb{C}^{\boldsymbol{K}\times\boldsymbol{M}}$, the transmitted signal is represented by $X = (x_1\ x_2 \ldots x_M)^T \in \mathbb{C}^{\boldsymbol{M}\times\boldsymbol{1}}$, and $N = (n_1\ n_2 \ldots n_K)^T \in \mathbb{C}^{\boldsymbol{K}\times\boldsymbol{1}}$ is the additive white Gaussian noise in the transmission. The operator $(\ldots)^T$ defines the transposed vector form.

Focusing on the 5G NR standard, before transmitting the signal, the information is built along the Downlink Shared Channel (DL-SCH) and Physical Downlink Shared Channel (PDSCH). The first stage involves error detection, segmentation, channel-coding operations based on Low-Density Parity-Check (LDPC) technique, rate-matching, and finally reconstructing the codeblocks into codewords [17]. The subsequent phase of the downlink chain is the PDSCH. The codeword first becomes subject to a scrambling operation and then undergoes symbol modulation. The symbols are mapped onto L layers. The number of layers determines how many independent streams can be transmitted in parallel, $S = (s_1\ s_2 \ldots s_L)^T$. Because every signal can be considered as the product of a linear combination, the data layers are not directly transmitted, but they are input into the precoding module. As a consequence, the input–output model can be rewritten as Equation (2).

$$Y = \boldsymbol{H}\, \boldsymbol{P} S + N \tag{2}$$

where $\boldsymbol{P} \in \mathbb{C}^{\boldsymbol{M}\times\boldsymbol{L}}$ identifies the precoding matrix and it is a function of the wireless channel $\boldsymbol{H}$, and $S \in \mathbb{C}^{\boldsymbol{L}\times\boldsymbol{1}}$ is composed of the L layers.

Therefore, the transmission can be improved in the form of increased achievable capacity and interference cancellation defining an appropriate precoding matrix [18]. The improvement depends on the technique selected to obtain the precoding matrix (e.g., Maximum-Ratio Transmission (MRT) Precoder, Zero Forcing (ZF) Precoder, and Diagonalization, but all of them require knowledge of the channel [19].

The described model is well-known to explain the input–output relationship in the system, and highlights the improvement given by channel knowledge and interference cancellation. However, in practice, the signals involved in the transmission are multidimensional, and every signal on each antenna is defined in the frequency and time domains, that is, subcarriers and symbols. This structure composes the resource grid where one resource element is defined as the intersection of one subcarrier with a symbol. Therefore, the real dimensions of the channel are given by the number of subcarriers (N_{sc}), the number of symbols (N_{symb}), and the antennas at the receiver (K) and transmitter (M).

To collect the channel knowledge, the reference signal is allocated in some dedicated resource elements across the transmitted grid. In the 5G NR standard, the reference signal

is the DM-RS. The DM-RS provides high flexibility in the transmission configuration to cater to different deployment scenarios. This is possible thanks to multiple configurations in the resource grid. A first classification can be done considering the mapping type in the time domain. For mapping type A, the first DM-RS is located in symbol 3 or 4 of the slot regardless of where the data transmission starts, while mapping type B places the reference signal in the first symbol of the data allocation. In case of rapid change of the channel, additional reference symbols are included in the resource grid in order to ensure an accurate channel estimation. It is possible to configure a double DM-RS structure which supports up to four reference-signal instances per slot. This flexibility is mirrored also in the frequency domain, where two different types of the reference signals can be configured. Type 1 consists of mapping up to four orthogonal signals using a single-symbol DM-RS and up to eight orthogonal reference signals using the double-symbol DM-RS. In type 2, the maximum allowed numbers are six and twelve, respectively. In order to ensure an accurate estimation of the resource grid in both domains for the channel, the two configurations can be combined [20]. Some of the possible combinations are illustrated in Figure 1.

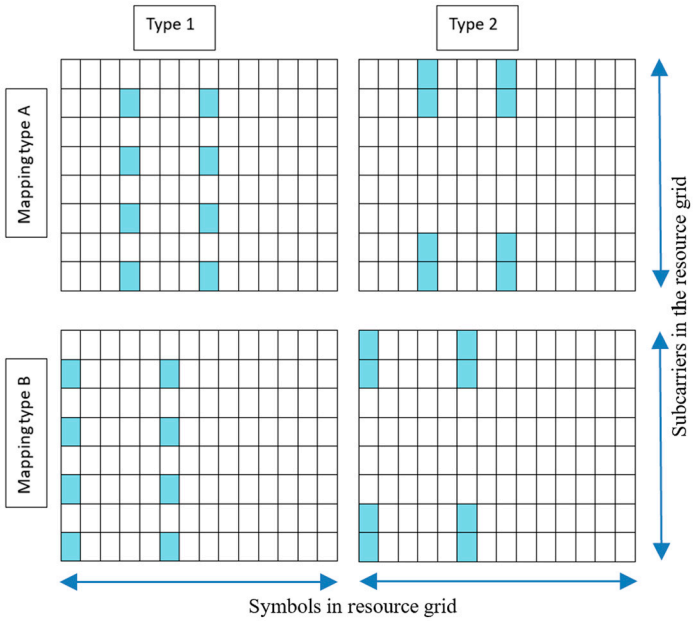

Figure 1. Example of DM-RS location in the resource grid.

At the receiver, the DM-RS location in frequency and time is communicated by the Downlink Control Information (DCI). The comparison between the received reference signal and the known reference signal generates the channel knowledge for specific resource elements. Using the channel estimation module, this knowledge is expanded and applied to the whole resource grid, generating the channel matrix. Despite the working principle being straightforward, the channel estimation is one of the most complex modules to be implemented due to variability of the channel over time and frequency [21].

3. Traditional Channel Estimation Methods

The most common channel estimation techniques are LS and MMSE. Both of these methods apply mathematical expressions to compute a closed form equation of the channel matrix using different cost functions [2]. The aim of the LS estimator is to minimize the square distance between the received and the reference signal, and uses Equation (3).

$$J_{LS}\left(\hat{\boldsymbol{H}}\right) = \|Y - X\hat{\boldsymbol{H}}\|^2 \tag{3}$$

It leads to the multiplication of the pseudo-inverse of the transmitted signal, $X \in \mathbb{C}^{M\times 1}$, by the received signal, $Y \in \mathbb{C}^{K\times 1}$, as in Equation (4).

$$\hat{H}_{LS} = (X^H X)^{-1} X^H Y \tag{4}$$

where the term X^H is the transposed transmitted signal. The outcome is the estimated channel for each subcarrier, $\hat{H}_{LS} \in \mathbb{C}^{K\times M}$.

Once the channel for specific elements in the resource grid is calculated, the information is interpolated in both the frequency and time domains to gain the whole channel knowledge. This technique is the most convenient in terms of complexity, and for this reason, it is still widely used. However, the accuracy of this method is strongly related to the signal-to-noise ratio (SNR) at the receiver [2,21]. In order to reduce the error and consider the degradation due to noise and spatial correlation, the MMSE includes the statistical information in the channel estimation.

In particular, MMSE aims is to minimize the error between the channel and the LS estimation, expressed by Equation (5).

$$J_{MMSE}(\hat{H}) = E\left\{ \|H - \hat{H}_{LS}\|^2 \right\} \tag{5}$$

This leads to the closed form Equation (6) for channel estimation.

$$\hat{H}_{MMSE} = R_{H\hat{H}} \left(R_{HH} + \frac{\sigma_n^2}{\sigma_x^2} I \right)^{-1} \hat{H}_{LS} \tag{6}$$

where $R_{H\hat{H}} \in \mathbb{C}^{K\times K}$ is the cross-correlation matrix between the exact channel and the temporary channel estimated in the frequency domain, while the R_{HH} is the autocorrelation of the exact channel. The terms σ_n^2 and σ_x^2 take into account the variance of the noise and the transmitted signal, respectively. The achievable performance is much higher than the LS technique. However, the second-order statistical parameters are highly demanding in terms of computational cost, and they are hardly accurate in a real system with a fast-varying environment [2,21,22].

4. DL-Channel Estimation

As detailed in the previous section, the LS and MMSE are the most common solutions for channel estimation, but both have associated limitations: LS lacks accuracy, while MMSE has high complexity. In light of hardware implementation, both aspects are critical. Additionally, the wireless input–output relationship could be affected by non-linear phenomena, which could potentially diminish the effectiveness of conventional methods [23].

In this vision, neural networks can learn quickly how to deal with different environments being unchained by mathematical equations. This is achievable because a large dataset can be generated simulating all the possible scenarios, and the neural network learns through offline training.

The classic neural network is able to learn thanks to the neurons present at each layer. Indeed, generally, the classic DNN consists of multiple layers, and at each layer, new useful data can be extrapolated. The gained information across layers depends on the type of DNN used. The most promising neural network structures for the channel estimation application presented in the literature are the FCDNN, the CNN, and the LSTM cells neural network. The LSTM is particularly popular in wireless applications because it is capable of building long-term memory associated with the data. Generally, these consist of fully connected layers composed by LSTM cells, as shown in Figure 2.

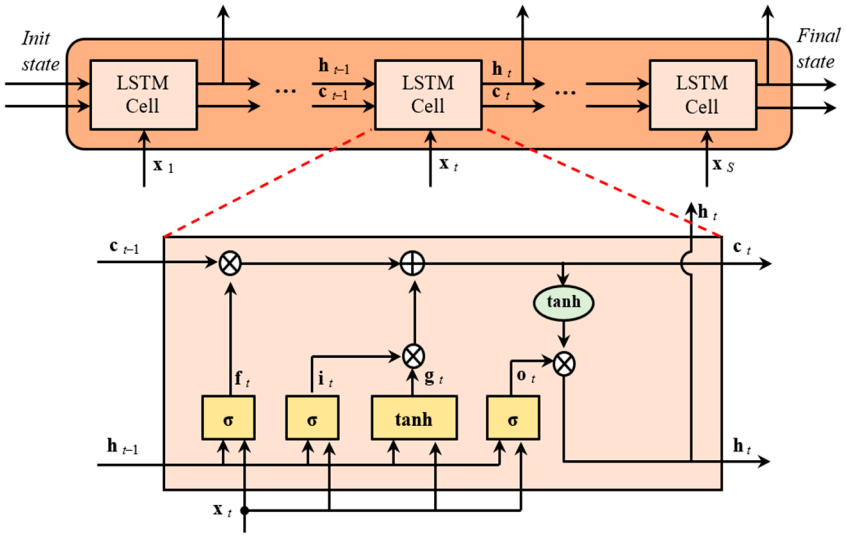

Figure 2. The LSTM layer.

Each LSTM cell has neurons which apply sigmoids or the hyperbolic tangents. Due to the complex functions in the neurons, a hardware implementation is challenging. Instead, more hardware-friendly functions are applied in the FCDNN and CNN neurons. The core function of the neuron in FCDNN is a weighted-sum, and every neuron of each layer is mapped to the subsequential layer, as shown in Figure 3 [24].

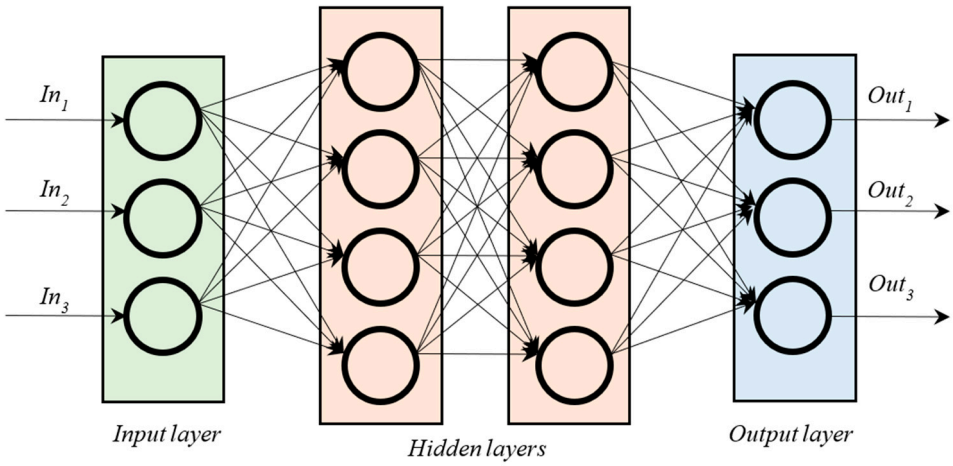

Figure 3. Example of FCDNN with two hidden layers.

Instead, the core function of the CNN is a 3-D Multiply-Accumulation (MACs) and its layers are sparsely connected. Given the dense connectivity among the layers, FCDNNs result in high complexity and memory requirements to store the needed parameters (i.e., weights and biases). When FCDNNs must either process high-dimensional data or consist of many hidden layers, their complexity becomes problematic. This issue is overcome by the CNN truncating some connections, thus alleviating the amount of operations to be performed. In addition, when some parameters are shared among different connections, the memory demand is lowered. This works well with images, where the locality of data is essential to extract useful features/relationships from pixels.

The CNN is able to extract and move local information from one layer to another using multiple feature maps (*fmaps*).

Consequently, the dimension of the current layer depends on the features of the previous layer, where the planar section is defined with the height (M_H) and width (M_W) of the previously extracted *fmaps*, and the depth is given by the number of filters (M_C), as shown in Figure 4.

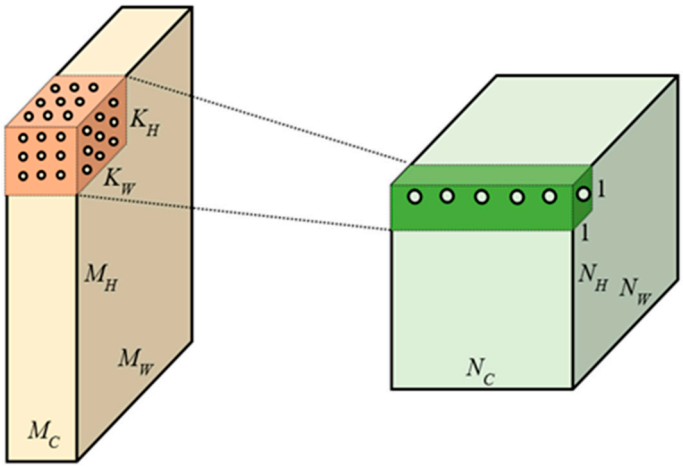

Figure 4. Example of a convolutional layer having $M_C = 3$, $K_H = K_W = 3$, $N_C = 6$. The yellow volume refers to the input neurons, while the light green volume refers to the output neurons. Accordingly, the orange block is the input sliding window, while the green block indicates the output neurons interested by that window.

At each layer, the filters extract multiple features from the input by spanning the current *fmap* with a sliding window. This window comprises $K_H \times K_W$ neurons, which are associated with learnable weights, and each window performs convolutions in the local region of the current *fmap*. At the end of this process, the outcome of each filter corresponds to a new *fmap* ($N_H \times N_W$). The set of all the calculated *fmaps* in the current layer is going to be the depth of the next layer (N_C). Once all the *fmaps* have been processed, the planar size generated for each CONV layer is given by the following Equation (7)

$$N_{H(W)} = \frac{M_{H(W)} - K_{H(W)} + 2P_{H(W)}}{S_{H(W)}} + 1 \tag{7}$$

where $P_{H(W)}$ is the row (column) padding size and $S_{H(W)}$ is the row (column) stride. The padding size indicates the number of additional rows and columns added at the borders of the input *fmaps* (e.g., by using zero values) to extend the sizes of the output *fmaps*. The stride refers to the distance between adjacent sliding windows.

Proposed Convolutional Neural Network

Following the previous considerations among the pertinent neural network models, the need for lower computational cost and simpler operations suggests focusing on the convolutional neural network. However, the configuration of a CNN structure can make a substantial difference, as there are several degrees of freedom that can be explored before disqualifying a specific structure, since the complexity and the accuracy of a neural network varies with the parameters of a specific application. For example, a neural network can have a simple structure, but this could be highly dependent on the dimension of the input, and to achieve a certain level of accuracy, the input size has to increase. Therefore, a more complex neural network, in which computational cost varies less with the input, could be the preferred choice. In light of the above, the choice of the CNN structure is not enough to guarantee favorable characteristics, but all the parameters involved in the system play a significant role. In the literature, the DL-channel estimation is used to improve the outcomes of the LS technique. In [3,8], this resulted in performing the LS calculations only on the reference signal and applying the neural network on the outcome. This increases the accuracy of the estimated channel in a few subcarriers generating a partial outcome. The latter is, then, interpolated across the subcarriers to obtain the full channel matrix. In other designs (e.g., [3,8,11]), the interpolation is a preliminary or middle stage of the channel estimation process using the neural network.

Unlike previous art, our proposal exploits the CNN to: (a) assist the channel estimation of a preliminary LS stage on the reference symbols only; (b) avoid the interpolation stage; (c) completely leverage only one CNN for the estimation of the full channel matrix. The proposed architecture is illustrated in Figure 5.

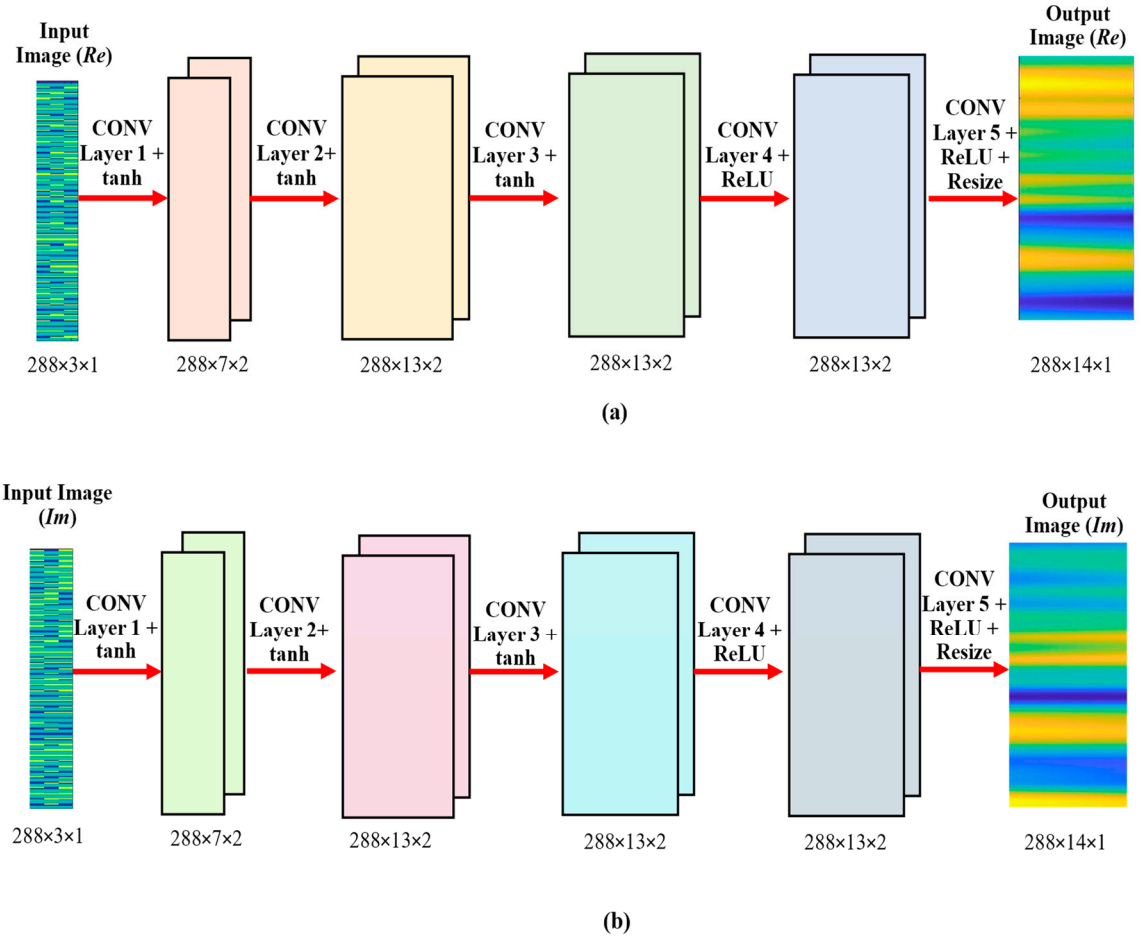

Figure 5. Diagram of the proposed neural network. The real (**a**) and imaginary (**b**) parts are processed in parallel.

The estimation of the channel matrix, **H**, is arranged as a 2-D image having N_{SC} rows and N_{OFDM} columns, which are the number of subcarriers and symbols, respectively. Since each value is a complex number, it is split into its real and imaginary parts, thus composing two different images, which are provided to the CNN in a parallel manner. In this application, the full channel grid is denoted with N_{SC} = 288 and N_{OFDM} = 14, but only the DMRS resource elements are provided as input to the network (N_{SC} = 288, N_{OFDM} = 3).

In the first layer, the symmetric padding is used to increase the horizontal size from 3 to 7. The same process is applied in the second and fifth layers, while the third and the fourth layers are classic convolutional layers, and the output has the same dimension of the input. An activation function follows each layer: the hyperbolic tangent (tanh) is applied after the first three layers, while the Rectified Linear Unit (ReLU) is used after the fourth and fifth layers. At the end, this neural network structure provides the full channel matrix as an output. To the best knowledge of the authors, this is the first time where the CNN is used to increase the resolution of the input without additional processing. Another novelty in the structure is related to the number of channels per layer. Considering the simplicity of an image which characterizes the estimated channel, the enormous amount of channels per layer proposed in the literature is not needed. This can show signs of overfitting in the networks, especially when the amount of training data is limited. Additionally, fewer

channels translate in less data to store and process, which reduces the memory requirements and quickens the training process.

Every design choice of the proposed CNN is a strategic decision aimed at balancing the model's performance, computational efficiency, and resource constraints for practical real-world applications. This was achieved by a combination of empirical testing and insights from relevant prior research. Each parameter involved in the neural network, such as dimension of the kernel, padding, and number of channels, was tuned at each layer. This allowed us to notice that it is more advantageous in term of accuracy to include larger kernels in the first layers rather than later ones. Instead, the number of channels was identified as a parameter that could be minimized to reduce computational costs. This is due to the fact that the channel, seen as an image, does not have highly detailed information.

A similar investigation was conducted for the padding method and the location of the size increments. Three different types of padding methods were considered: replication of the edges, zero-padding, and symmetric padding. The latter result was more effective thanks to propagation of the DMRS structure in the resource grid, while the location of the size increment was found to be more effective when placed at the beginning of the network.

In order to assess the advantages of this model, two additional CNN structures with pre-interpolation stages have been implemented. These two architectures are going to differ in the number of layers; one is composed of three layers while the other is designed with six layers. Increasing the number of layers should increase the accuracy, compromising the computational cost, giving good benchmarks in the subsequential analysis. An illustrative comparison is also included in Figure 6, where Figure 6a describes the traditional approach using interpolation, and Figure 6b shows in detail all of the settings at each layer in the proposed CNN.

The model reported in Figure 6a receives the preliminary estimation provided by LS and the results are interpolated, increasing the size of the input from 288×3 to 288×14. The Input Image Layer forwards the input image to the first CONV Layer, which adopts 3-D filters with planar sizes of 9×9. Padding sizes are set to $P_H = P_W = 4$ in order to preserve the sizes of *fmaps*. A sequence of L–2 CONV Layers, with L being the number of layers, is stacked throughout the model. Each of them consists of several filters with 5×5 planar sizes. Moreover, in this case, the sizes of the generated *fmaps* are preserved by setting $P_H = P_W = 2$. Finally, the last CONV Layer performs the remaining computations to generate the estimated channel. It is worth underlining that the stride $S_H = S_W = 1$ throughout the CNN and the Rectified Linear Unit are used as non-linear activation.

The model reported in Figure 6b receives the original symbols after application of LS without an interpolation stage (288×3). The Input Image Layer forwards the input image to the first CONV Layer, which adopts 3-D filters having planar sizes 5×5. In this scenario, CONV layers are responsible for progressively increasing the size of input symbols to build the full-size channel grid. To reach this goal, the padding sizes are tuned in each layer. In this case, $P_H = 2$ allows the number of subcarriers to be kept unchanged, while $P_W = 4$ allows the size N_{OFDM} to grow from 3 to 7. The second CONV layer adopts 9×9 filters, which lead to 288×13 planar representations. The last three CONV layers, having $K_H = K_W = 5$, extract further features from internal data and specifically the fifth layer increases the size from 13 to 15. At the end, a Resize layer is used to crop 1 column to meet the desired number of OFDM symbols (288×14).

In order to robustly estimate the channel and adjust the weights of the network for this application, the CNNs are trained offline using an image-to-image regression. At each iteration, the training data are generated varying the channel profile, Doppler effect, delay spread, and SNR value, in order to consider a wide range of environmental 5G scenarios. Each value associated to each parameter is randomly selected from the predefined valid range, as shown in Table 1.

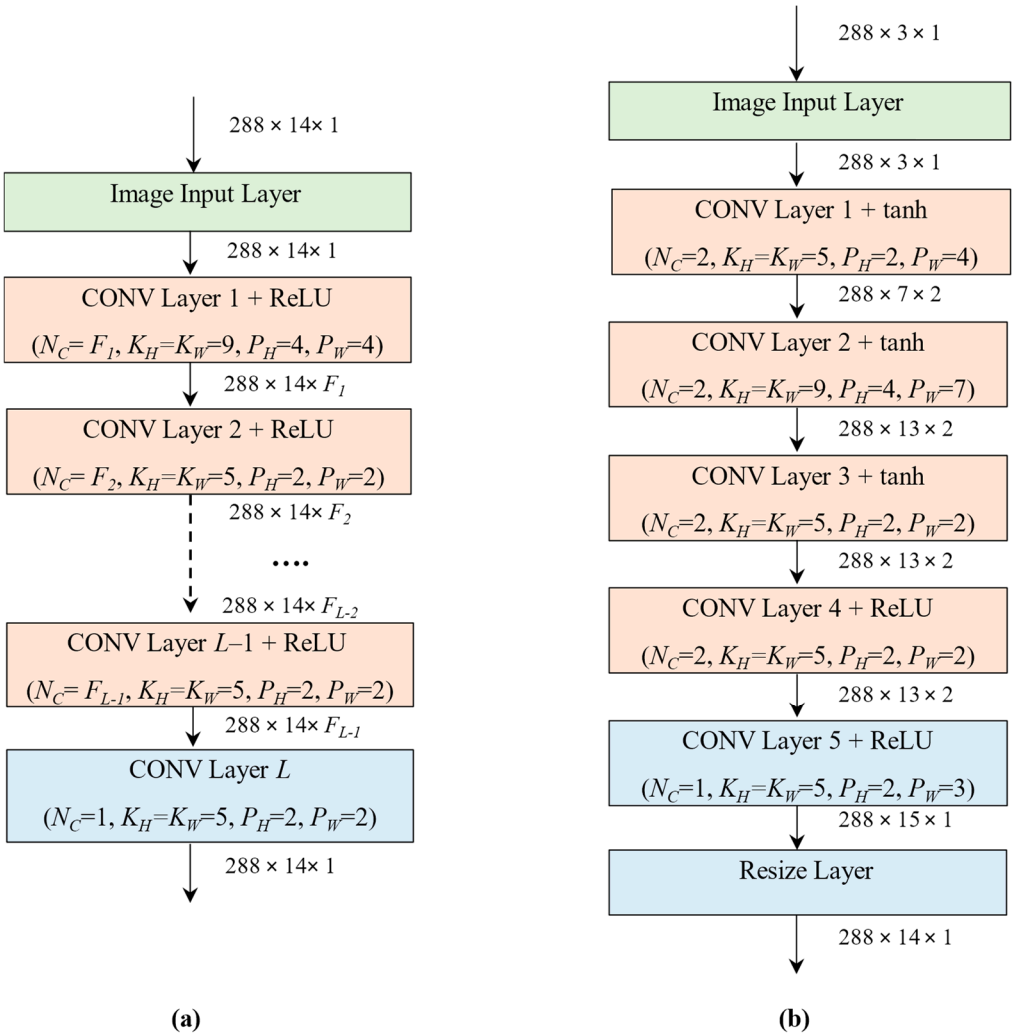

Figure 6. Comparison of the implemented CNNs: (**a**) the model of CNN that enhances the estimation provided by a preliminary interpolation stage; (**b**) the model of the proposed neural network that completely estimates the channel.

Table 1. Configurable parameters in the neural network training.

Parameters	Range
Channel profiles	TDL-A, TDL-B, TDL-C, TDL-D, TDL-E
Doppler effect	20–200 Hz
Delay spread	100–300 ns
SNR levels	−5 to 15 dB

For each iteration of training, the predicted image is compared to the perfect channel estimation using the loss function reported in Equation (8).

$$loss = \frac{1}{2} \sum_{i=1}^{O_H \times O_W \times O_C} (\boldsymbol{H_i} - \boldsymbol{Hp_i})^2 \quad (8)$$

where O_H, O_W, and O_C, represent the height, width, and depth of the channel response, respectively. The target image $\boldsymbol{H}$ is the actual channel, while the prediction $\boldsymbol{Hp}$ is the estimated channel. The aim of the training is to minimize such loss by progressively updating the parameters of the CNN.

5. Computational Cost

Towards an efficient hardware implementation, the computational cost is a fundamental criterion to analyze. With offline training, the complexity depends strongly on the neural network dimension and topology. Because the design choices largely affect the number of operations involved in the computation, an analysis among the state-of-the-art DL-channel estimation techniques is presented. When considering the computational cost, the primary focus lies in identifying the dominant factors that contribute the most to the overall growth rate of the algorithm's complexity. This helps to simplify the calculation, focusing on the most significant operations.

The steps of this analysis are summarized in Tables 2 and 3, where each column highlights an additional stage before reaching the total computational cost.

Table 2. Complexity derivation for traditional methods per stage.

Channel Estimation Techniques	Complexity per Reference Elements in Symbol (C_{rElem})	Complexity per Reference Symbols (C_{rs})	Complexity of the Interpolation	Complexity per Tx and Rx Antennas
MMSE	$C_{rElem} = N^3_{RefElem} + 3\,N^2_{RefElem} + 3\,N_{RefElem}$ [25]	$C_{rs} = C_{rElem}\,N_{RefSymb}$	$C_{interp} = C_{rs} + 3\,N_{noRefElSlot}$	$C_{MMSE} = C_{interp}\,N_{TxAnts}N_{RxAnts}$
LS	$C_{rElem} = 4\,N_{RefElem}$ [25]	$C_{rs} = C_{rElem}\,N_{RefSymb}$	$C_{interp} = C_{rs} + 3\,N_{noRefElSlot}$	$C_{LS} = C_{interp}\,N_{TxAnts}N_{RxAnts}$

Table 3. Complexity derivation for DL-based channel estimations per stage.

Channel Estimation Techniques	Complexity of the Input Computation	Complexity of the DL Method per Layer	Complexity of the Post-Interpolation ($C_{intpost}$)	Total Complexity
CNN-L3	$C_{interp} = C_{rs-LS} + 3\,N_{noRefElSlot}$	$C_{CNN} = N_C\,M_C K_H K_W N_H N_W$	N/A	$C_{tot} = (C_{interp} + 3\,C_{CNN})\,N_{TxAnts}N_{RxAnts}$
CNN-L6	$C_{interp} = C_{rs-LS} + 3\,N_{noRefElSlot}$	$C_{CNN} = N_C\,M_C K_H K_W N_H N_W$	N/A	$C_{tot} = (C_{interp} + 6\,C_{CNN})\,N_{TxAnts}N_{RxAnts}$
FCDNN [3]	$C_{LS-interpInFreq} = (C_{rs-LS} + 3\,N_{noRefElSymb})\,N_{TxAnts}N_{RxAnts}$	$C_{FCDNN} = N_{neur}\,M_{input} + N_{neur}$ where $M_{input} = N_{TxAnts}N_{RxAnts}$	$C_{intpost} = 3\,N_{noRefElSlot}$	$C_{tot} = C_{LS-interpInFreq} + N_{RefSymb}(4\,C_{FCDNN}) + C_{intpost}$
CNN [8]	$C_{LS-interpInFreq} = (C_{rs-LS} + 3\,N_{noRefElSymb})\,N_{TxAnts}N_{RxAnts}$	$C_{CNN} = N_C\,M_C K_H K_W N_H N_W$ and $C_{FCDNN} = N_{neur}\,M_{input} + N_{neur}$	$C_{intpost} = 3\,N_{noRefElSlot}$	$C_{tot} = C_{LS-interpInFreq} + N_{RefSymb}(4\,C_{CNN} + 1\,C_{FCDNN}) + C_{intpost}$
LSTM [8]	$C_{LS-interpInFreq} = (C_{rs-LS} + 3\,N_{noRefElSymb})\,N_{TxAnts}N_{RxAnts}$	$C_{LSTM} = 4N_s\,D_{HL}(2N_{feat} + 2D_{HL} + 1)$ and $C_{FCDNN} = N_{neur}\,M_{input} + N_{neur}$	$C_{intpost} = 3\,N_{noRefElSlot}$	$C_{tot} = C_{LS-interpInFreq} + N_{RefSymb}(2\,C_{LSTM} + 1\,C_{FCDNN}) + C_{intpost}$
Proposed CNN	C_{rs-LS}	$C_{CNN} = N_C\,M_C K_H K_W N_H N_W$	N/A	$C_{tot} = 4\,C_{CNN}\,N_{TxAnts}N_{RxAnts}$

$N_{RefElem}$ is the number of the reference resource elements in the symbol. $K_H K_W$ are the dimensions of the used kernel in the CNN. $N_{RefSymb}$ is the number of the reference symbols in a slot. $N_H N_W$ are the dimensions of the CNN current layer output. $N_{noRefElSlot}$ is the number of the resource elements not containing a reference resource element in the slot. N_{neur} is the number of neurons in the FCDNN current layer. $N_{noRefElSymb}$ is the number of the resource elements not containing a reference resource element in the symbol. D_{HL} is the size of the LSTM hidden layer. $N_{TxAnts}N_{RxAnts}$ are the number of antennas at the Tx and at the Rx. N_s is the number of LSTM cells. $N_C\ M_C$ are the number of channels in the CNN current layer and the subsequential one. N_{feat} is the number of LSTM features.

The considered total computation is the complexity needed to estimate the full channel ($N_{sc} \times N_{symb} \times K \times M$) from the reference elements in the grid. The first table includes the computation process for LS and MMSE techniques in order to provide a benchmark against the traditional channel associated limitations: LS lacks in accuracy, while MMSE has high complexity. The second table provides all the information per stage to compute the complexity of the analyzed DL-based channel estimation methods.

This helps to highlight the possible reduction in complexity avoiding some stages. Additionally, in the prospect of massive MIMO application, a useful prediction of the complexity growth is given by the increased number of antennas at one end. Figures 7 and 8 show the total number of operations required to calculate the full channel matrix from the pilot signals for different numbers of antennas at the receiver for each solution.

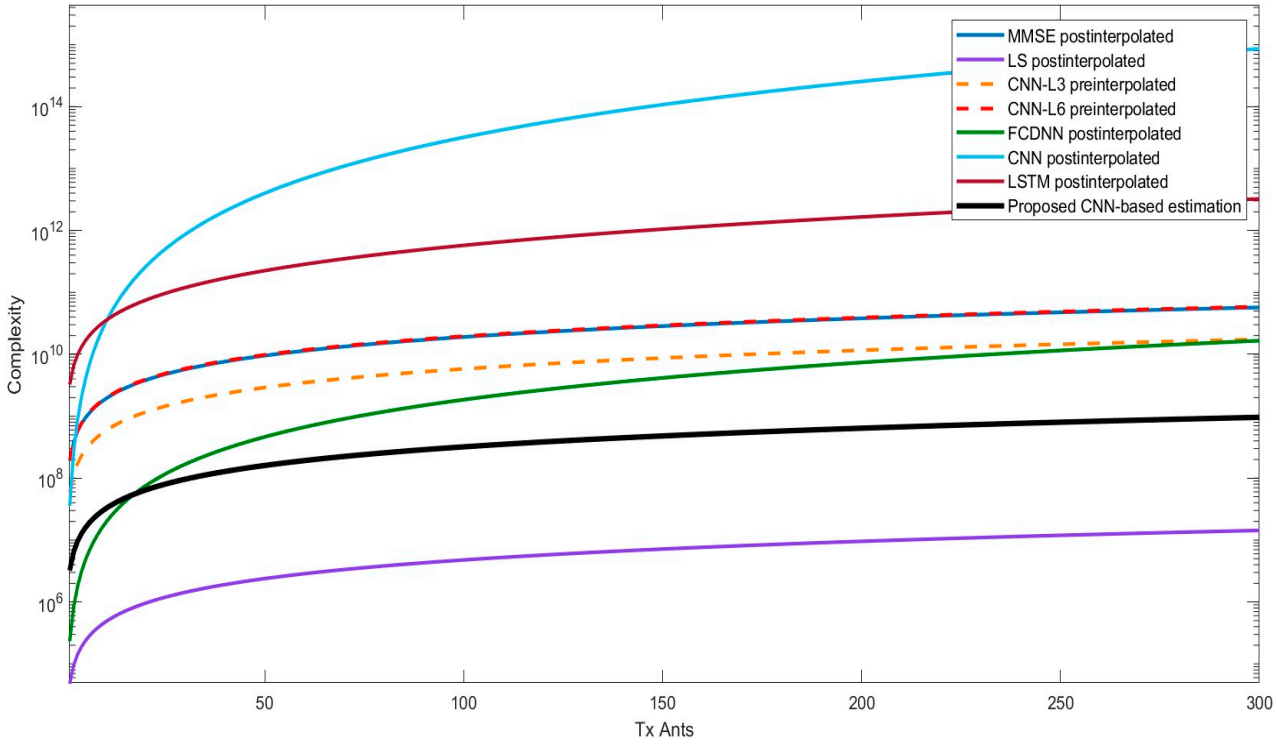

Figure 7. Computational cost analysis among several DL channel estimations varying the number of antennas at the receiver. The transmitter (user terminal, UT) uses two antennas. The calculation is obtained from the above table considering the implemented designs and those reported in the literature [3,8,25].

The MMSE and LS techniques represent the two benchmarks for evaluating the feasibility on hardware of the rest of the techniques. Being the MMSE traditionally discarded for implementation, clearly, the CNN network presented in [8] is not suitable, as the complexity is 2 orders of magnitude higher than the MMSE. This highlights that CNN-based structure, where the number of neurons in the network is not dependent on the number of antennas, have a completely different order of complexity, indicating that the required computational cost varies with design choices.

Indeed, the implemented CNNs with pre-interpolation are comparable to the MMSE complexity. Although there is a big difference in number of layers, the computational cost varies by half an order of magnitude. In this range, the FCDNN and LSTM networks presented in [3,8] are included. The growth of the FCDNN complexity is faster than the LSTM, having the number of neurons per layer dependent on the number of antennas in the system.

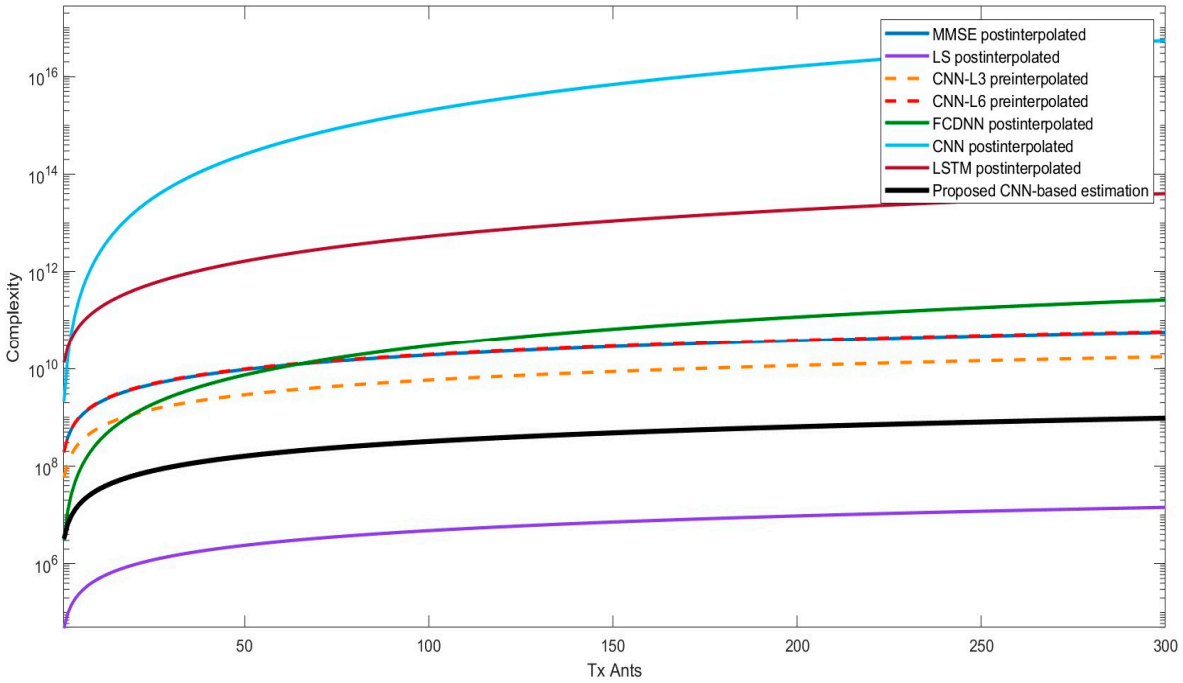

Figure 8. Computational cost analysis among several implemented DL channel estimations varying the number of antennas at the receiver. The transmitting UT uses eight antennas. The calculation is obtained from the above table considering the implemented designs and those reported in the literature [3,8,25].

As expected, the LS channel estimation with interpolation in frequency and time domains is traditionally the least expensive in terms of computational cost. However, our proposal provides the next lowest complexity, outperforming all other DL-based techniques. The improvement results in terms of complexity are due to the full channel being obtained as the output of the neural network, completely avoiding the interpolation stage and a hardware-oriented CNN design.

The other three important factors in the choice of the hardware-oriented neural network when it comes to their implementation are the memory usage, the opportunities offered for parallelization, and the type of required operations. Because of the use of locality, the CNN requires less memory for storing parameters, and each local section can be parallelized. The other advantage is related to the type of involved operations since convolution is predominant. In the FCDNNs, each neuron is connected to the neuron in the subsequent layer. Therefore, the generated information has to be stored. This is true also for the LSTM because it captures a recurrent property over time. As a consequence, data have to be stored and propagated to the next cell to obtain the final result, therefore each result is dependent on the previous one. Additionally, the latter uses sigmoids and hyperbolic tangents in each cell, which are not hardware-friendly operations.

In conclusion, the proposed technique outperforms other DL alternatives when it comes to implementation. In the next section, we evaluate the performance in terms of accuracy in the channel estimation. Since FCDNN and LSTM present unfavorable characteristics when it comes to their hardware implementation, the focus will be oriented to the CNN structures. The LSTM is excluded, while the comparison with the FCDNN structure is purely based on the neural network presented in [3].

6. Numerical Simulation and Analysis

To evaluate the performance of the proposed CNN-channel estimation in terms of accuracy in the channel matrix estimation, the MSE is analyzed over a range of SNR values. The main setup parameters are summarized in Table 4, and the simulated system is shown in Figure 9.

Table 4. Main setup parameters for the simulated system.

Parameters	Settings
Antennas	2 Rx and 2 Tx
Resource blocks	24
Subcarrier spacing	15
Mapping type	A
Layer	1
Symbol Allocation	[0 14]
DMRS configuration	DMRS Configuration type = 2 DMRS length = 1 DMRS Additional Position = 2
Nfft	512
Channel	Channel profile = TDL-A Doppler Shift = 36 Delay spread = 300×10^{-9}

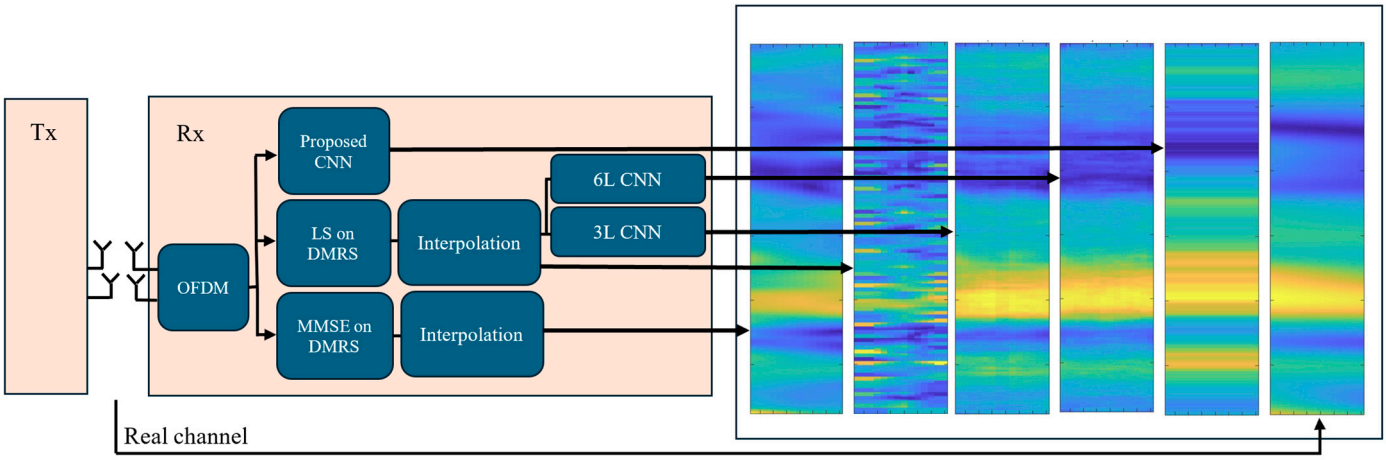

Figure 9. Diagram of the simulated system comparing the channel estimation from the proposed structure with CNNs with pre-interpolation, LS with post-interpolation, and MMSE with post-interpolation.

Two antennas for the receiver and the transmitter are considered. The transmitted grid is generated considering only the DMRS signals in the PDSCH.

At the receiver, the channel information is estimated using five different types of channel estimation designs and compared with the perfect knowledge of the full channel. These are the conventional LS estimation, the MMSE estimation, the pre-interpolated CNN channel estimation with 3L and 6L, and an FCDNN-based channel estimation proposed in [3]. The MSE results are obtained using MATLAB's 5G Toolbox [26] and are presented in the logarithmic scale in Figure 10.

All of the MSE curves improve as the SNR increases, following a linear trend. The difference is in the rate of decreasing MSE. The LS technique decreases faster with the SNR growth, while the rest of the curves present a slower trend. However, as expected, the traditional LS technique generates the highest MSE, becoming competitive only at high SNR values. On the other hand, the neural networks (CNN-6L, CNN-3L, FCDNN, and the proposed CNN-based channel estimation) are all very close to the MMSE technique, commonly considered the most accurate. The lowest MSE values are obtained by the technique described in [3] and the CNN-6L design. However, they are not convenient in terms of complexity considering their dependency with the number of antennas and the evolution of this parameter in the communication systems (e.g., Massive MIMO and Gigantic MIMO [27]), especially for the FCDNN network where the number of neurons is proportional to the total number of antennas in the system.

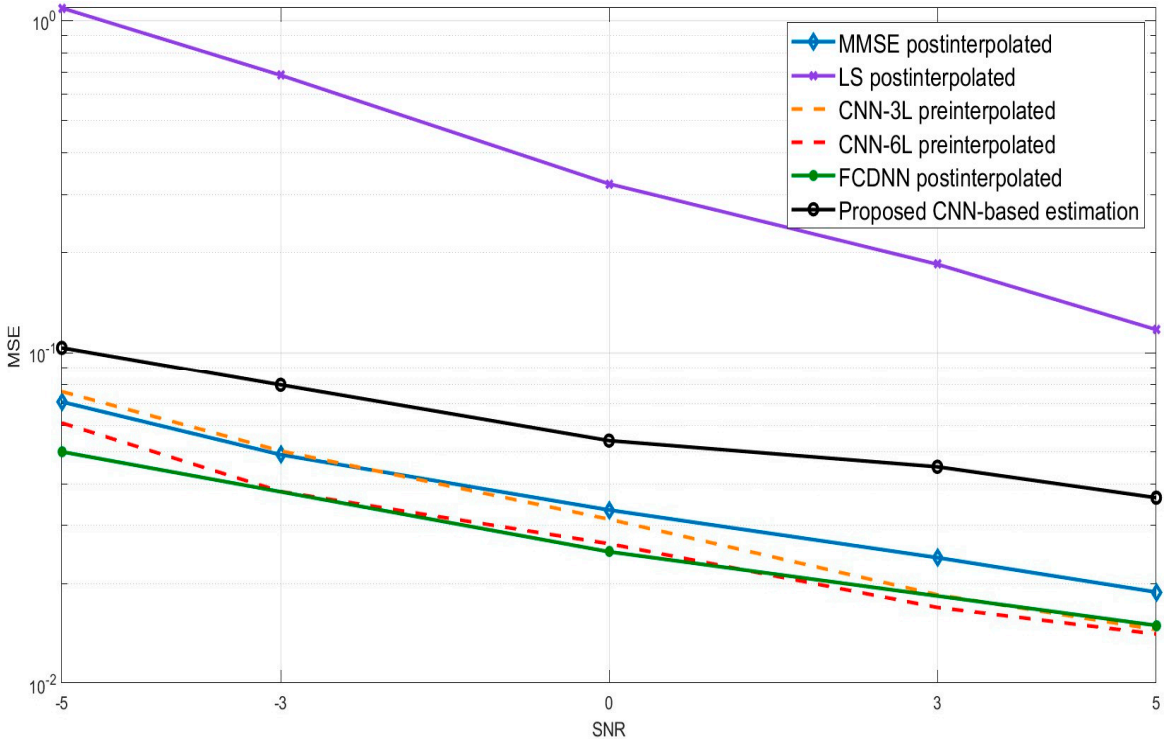

Figure 10. MSE analysis varying the SNR. The comparison is between the implemented designs and the FCDNN postinterpolated proposed in [3].

Despite a slightly reduced accuracy achieved by the proposed technique, its overall performance is very much in line with the state-of-the-art methodologies. This reduction is expressed in an increase of the MSE at each SNR level, and it is expected due to the approximation introduced for reducing the complexity. The first factor which played an important role is the use of the padding method to increase the dimensions in favor of the computational cost. Therefore, the interpolation stage is not needed, and the full-channel dimensions are achieved symmetrically mirroring the reference information. Two other important factors in the reduction of accuracy are the low number of channels and layers in the proposed CNN. As already mentioned, each layer helps to denoise the input and the channels collect the feature information. Indeed, the impact of increasing the number of layers and channels is also proven by the CNN-6L and CNN-3L neural network performances. Increasing the number of layers appears especially beneficial at lower SNR levels, though its impact diminishes at higher SNR values.

Despite these hardware-oriented design choices impacting the MSE analysis, the proposed CNN outperforms the LS technique by an order of magnitude at low SNR levels and maintains competitive performance at higher SNRs. The largest deviation from MMSE is 0.018 at 5 dB, which is minimal, concerning the ability of the transmission system to recover the information using up-sampling and error correction post-channel estimation [3]. Therefore, with its hardware implementation benefits, this CNN-based channel estimation approach thus proves to be a viable option for 5G NR systems, showing that effective results can be achieved through neural networks without requiring highly complex designs.

7. Conclusions

Every channel estimation method based on neural networks found in the literature needs multiple stages to estimate the whole channel. In this paper, we presented a convolutional neural network for channel estimation which has the LS applied at DMRS signals as an input and generates the whole channel as an output. With the purpose of a future hardware implementation, the performance of the proposed CNN is evaluated using two terms of comparison: computational cost and the mean square error. Our CNN has the

lowest complexity among the analyzed neural networks such as FCDNN, LSTM, and other CNN architectures. Additionally, Tables 2 and 3 present the mathematical expressions to compute the complexity, not only for the neural networks included in this study, but with the computational cost divided by stages, for any model which uses these types of layers. Moreover, the results of the error analysis prove competitive MSE performance to the MMSE channel estimation applied to a 2×2 MIMO system, having a maximum discrepancy of 0.018 at 5 dB SNR. On the basis of these analyses, the proposed CNN technique gives a deployable solution outperforming earlier approaches reported in the literature. In this vision, a further study on the integration of the real and imaginary parts could lead to additional resource savings. Subsequently, quantizing the proposed CNN and implementing it on VERSAL technology can provide a definitive validation for the CNN-based channel estimation method.

Author Contributions: Formal analysis, S.S. and C.S.; funding acquisition, G.G.; investigation, S.S.; methodology, S.S. and C.S.; project administration, G.G.; supervision, G.G.; writing, S.S. and C.S.; review and editing, S.B.S., Y.D., S.S., C.S. and G.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kim, H.; Jiang, Y.; Rana, R.; Kannan, S.; Oh, S.; Viswanath, P. Communication Algorithms via Deep Learning. *arXiv* **2018**, arXiv:1805.09317.
2. Cho, Y.S.; Kim, J.; Yang, W.Y.; Kang, C.G. *MIMO-OFDM Wireless Communications with MATLAB*; John Wiley & Sons (Asia): Singapore, 2010.
3. Le Ha, A.; Van Chien, T.; Nguyen, T.H.; Choi, W.; Nguyen, V.D. Deep Learning-Aided 5G Channel Estimation. In Proceedings of the 2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM), Seoul, Republic of Korea, 4–6 January 2021; pp. 1–7. [CrossRef]
4. Chun, C.-J.; Kang, J.-M.; Kim, I.-M. Deep Learning-Based Channel Estimation for Massive MIMO Systems. *IEEE Wireless Commun. Lett.* **2019**, *8*, 1228–1231. [CrossRef]
5. Qin, Z.; Ye, H.; Li, G.Y.; Juang, B.-H.F. Deep Learning in Physical Layer Communications. *IEEE Wireless Commun.* **2019**, *26*, 93–99. [CrossRef]
6. Belgiovine, M.; Sankhe, K.; Bocanegra, C.; Roy, D.; Chowdhury, K.R. Deep Learning at the Edge for Channel Estimation in Beyond-5G Massive MIMO. *IEEE Wireless Commun.* **2021**, *28*, 19–25. [CrossRef]
7. Zhou, X.; Zhang, J.; Syu, C.-W.; Wen, C.-K.; Zhang, J.; Jin, S. Model-Driven Deep Learning-Based MIMO-OFDM Detector: Design, Simulation, and Experimental Results. *IEEE Trans. Commun.* **2022**, *70*, 5193–5207. [CrossRef]
8. Le Ha, A.; Van Chien, T.; Nguyen, T.H.; Choo, H.; Nguyen, V.D. Machine Learning-Based 5G-and-Beyond Channel Estimation for MIMO-OFDM Communication Systems. *Sensors* **2021**, *21*, 4861. [CrossRef] [PubMed]
9. Dai, T.; Cai, J.; Zhang, Y.; Xia, S.T.; Zhang, L. Second-Order Attention Network for Single Image Super-Resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 11065–11074.
10. AMD. Versal Architecture and Product Data Sheet: Overview. DS950 (v2.2). Available online: https://docs.amd.com/v/u/en-US/ds950-versal-overview (accessed on 4 June 2024).
11. Soltani, M.; Pourahmadi, V.; Mirzaei, A.; Sheikhzadeh, H. Deep Learning-Based Channel Estimation. *IEEE Commun. Lett.* **2019**, *23*, 652–655. [CrossRef]
12. Guo, J.; Chen, T.; Jin, S.; Li, G.Y.; Wang, X.; Hou, X. Deep Learning for Joint Channel Estimation and Feedback in Massive MIMO Systems. *Digit. Commun. Netw.* **2024**, *10*, 83–93. [CrossRef]
13. Mashhadi, M.B.; Gündüz, D. Pruning the Pilots: Deep Learning-Based Pilot Design and Channel Estimation for MIMO-OFDM Systems. *IEEE Trans. Wireless Commun.* **2021**, *20*, 6315–6328. [CrossRef]
14. Luan, D.; Thompson, J. Attention Based Neural Networks for Wireless Channel Estimation. In Proceedings of the 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), Helsinki, Finland, 19–22 June 2022; pp. 1–5. [CrossRef]
15. Khichar, S.; Santipach, W.; Wuttisittikulkij, L.; Parnianifard, A.; Chaudhary, S. Efficient Channel Estimation in OFDM Systems Using a Fast Super-Resolution CNN Model. *J. Sens. Actuator Netw.* **2024**, *13*, 55. [CrossRef]
16. Grami, A. *Introduction to Digital Communications*; Academic Press: Cambridge, MA, USA, 2015.

17. 3GPP. TS 38.212. NR; Multiplexing and Channel Coding. Version 16.1.0. 2020. Available online: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3214 (accessed on 26 May 2020).
18. Sanguinetti, L.; Björnson, E.; Hoydis, J. Toward Massive MIMO 2.0: Understanding Spatial Correlation, Interference Suppression, and Pilot Contamination. *IEEE Trans. Commun.* **2020**, *68*, 232–257. [CrossRef]
19. Luo, F.L.; Zhang, C.J. *Signal Processing for 5G: Algorithms and Implementations*; John Wiley & Sons: Chichester, UK, 2016.
20. Dahlman, E.; Parkvall, S.; Skold, J. *5G NR: The Next Generation Wireless Access Technology*; Academic Press: Cambridge, MA, USA, 2018.
21. Ozdemir, M.K.; Arslan, H. Channel Estimation for Wireless OFDM Systems. *IEEE Commun. Surv. Tutor.* **2007**, *9*, 18–48. [CrossRef]
22. Hu, Q.; Gao, F.; Zhang, H.; Jin, S.; Li, Y. Deep Learning for MIMO Channel Estimation: Interpretation, Performance, and Comparison. *arXiv* **2019**, arXiv:1911.01918.
23. Lv, C.; Luo, Z. Deep Learning for Channel Estimation in Physical Layer Wireless Communications: Fundamentals, Methods, and Challenges. *Electronics* **2023**, *12*, 4965. [CrossRef]
24. Sze, V.; Chen, Y.-H.; Yang, T.-J.; Emer, J.S. Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *Proc. IEEE* **2017**, *105*, 2295–2329. [CrossRef]
25. Liao, Y.; Hua, Y.; Cai, Y. Deep Learning Based Channel Estimation Algorithm for Fast Time-Varying MIMO-OFDM Systems. *IEEE Commun. Lett.* **2020**, *24*, 572–576. [CrossRef]
26. Zaidi, A.; Athley, F.; Medbo, J.; Gustavsson, U.; Durisi, G.; Chen, X. *5G Physical Layer: Principles, Models and Technology Components*; Academic Press: Cambridge, MA, USA, 2018.
27. Björnson, E.; Kara, F.; Kolomvakis, N.; Kosasih, A.; Ramezani, P.; Salman, M.B. Enabling 6G Performance in the Upper Mid-Band Through Gigantic MIMO. *arXiv* **2024**, arXiv:2407.05630.