CHANNEL ESTIMATION USING DEEP LEARNING TECHNIQUES

Thesis submitted by

Ritvik Kapila, Ritik Agrawal

2017EE10484, 2017EE10482

under the guidance of

Prof. Manav Bhatnagar

in partial fulfilment of the requirements for
the award of the degree of

Bachelor of Technology

Department Of Electrical Engineering

INDIAN INSTITUTE OF TECHNOLOGY DELHI

January 2021

THESIS CERTIFICATE

The thesis titled as **Channel Estimation Using Deep Learning Techniques**, submitted by **Ritik Agrawal, Ritvik Kapila**, to Indian Institute of Technology Delhi, for the award of the degree of **Bachelors of Technology**, is a bonafide record of the research work done by them under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof.Manav Bhatnagar
Professor
Deptt. of Electrical Engineering
IIT-Delhi

ACKNOWLEDGEMENTS

We are grateful to out faculty advisor as well as the project mentor, **Prof. Manav Bhatnagar**. He really pushed us to thrive and it has been nothing but a very important learning activity on our project - Channel Estimation using Deep Learning techniques. He really gave us a lot of insights and his knowledge in the field of communication is unparalleled.

ABSTRACT

In this thesis, channel estimation methods are discussed for an OFDM system. Conventional approaches such LS and MMSE are discussed along with their trade-offs with complexity and efficiency. Further, DL techniques such as SrCNN and DnCNN are exploited for Super Resolution and Denoising by considering the frequency-time response of channel matrix as a 2 Dimensional image. Finally a fusion is taken between a linear and non-linear network to come up with a decently complex and efficient model. The mean squared error or the mse between the actual received and the estimated signal is compared for various models to compute the model performance, The time complexity analysis is also done which is chosen as another performance criteria. The results are the testimony that the approach is indeed an efficacious one for channel estimation.

Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 Overview

In communications systems, wireless communication has emerged as one of most widely used modes of communication these days. In particular, selective frequency fading has been seen as a common problem as discussed in [7], [8] when multiple carrier frequencies are taken into account in wireless transmission. The modulation method OFDM or Orthogonal frequency-division multiplexing is widely used to address this problem and it has found its way in many modern day applications but majorly in 5G wireless communications. The transmitted signal though due to the scattering and the reflection at multiple paths becomes susceptible to delays and noise. Due to this reason the received signal becomes distorted and it becomes crucial to estimate the channel with all the imperfections modeled to maintain the communication link.

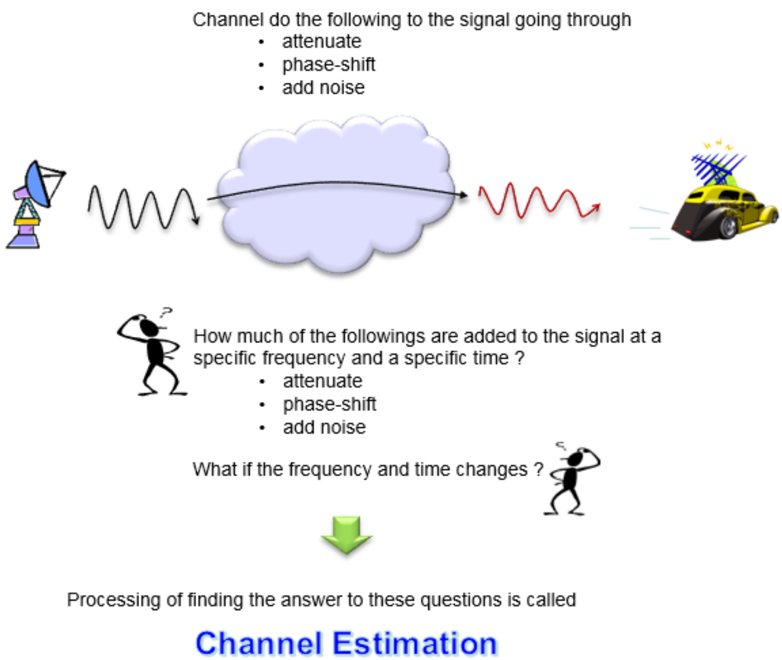

Figure 1.1: An illustration of Channel Estimation

The task of channel estimation involves reconstruction of channel with a reasonable accuracy at the pilot positions. The channel is thought of as a 2D image, where initially it is of lower resolution due to the limited number of pilot positions and the increasing the resolution via any channel estimation technique to reconstruct the complete channel. For this purpose some training is required and a system model is considered. In this research work, we shed light on some of the existing techniques and the proposed framework that provides a reasonable accuracy.

1.2 Motivation

In wireless communication, the channel portrays how a sign proliferates from the transmitter to the receiver. It speaks to the joint impact of, for instance, dispersing, blurring, and power rot with distance. Flawed estimate of the channel may bring about lessening, bending, postponements, and stage move of the showing up signs. In this way, channel estimation is needed to make up for the mutilation presented in the images, as they travel through the channel, and to consider SNR. Subsequently, channel estimation is a critical issue in wireless communication. The typical channel model for a system is as follows,

$$y = Hx + n \tag{1.1}$$

where,

- **y:** Received Signal
- **x:** Transmitted Signal
- **H:** Channel Matrix
- **n:** Noise Matrix, where the noise is often modeled as follows,

$$n \sim \mathcal{CN}(0,\, S) \tag{1.2}$$

Many techniques have been developed and are still being looked into for the task of estimating the channel estimation in communication systems. There are two approaches that are mainly used, the first one is the data aided approach and the second one uses blind estimation. The blind estimation method is blind to the training data from the transmitter side and only relies on the received data whereas the data aided approach uses some training data known by both the receiver and the transmitter. Since the latter uses more data or overhead than the former, it's accuracy is also better. Hence, a trade off lies between the two. But with some assumptions and modifications in the data aided approach, it becomes ideal for the estimation task.

The data aided approach in [6] or the pilot estimation method depending upon the knowledge of channel statistics beforehand is further classified in three t ypes: completely known channel statistics, partially known channel statistics and unknown channel statistics.

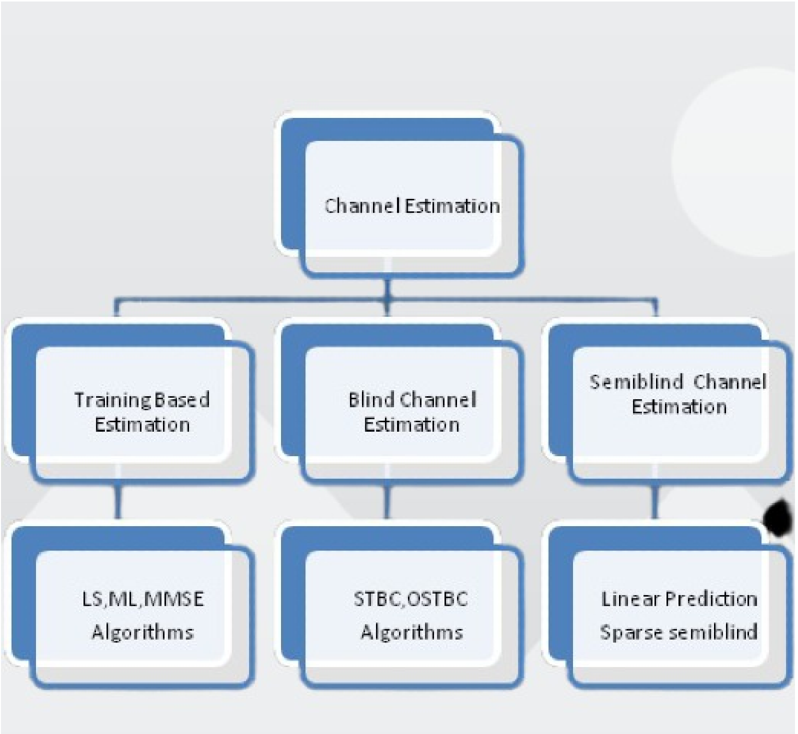

Figure 1.2: Various Techniques for Channel Estimation

We will look at some of the approaches in detail and decide which one performs the best with the standard benchmarks of the channel estimation task.

1.3 Objective

During the course of the project, there were two broad objectives that needed to be achieved. Since Deep Learning systems consider any typical wireless system as a black-box without taking the help of any prior knowledge as well as reducing the burden of channel statistics information, they become well suited candidates for the job. Also, there recent breakthroughs in the image processing and computer vision allows them to study channel matrix as well because of the notion that a typical channel matrix can be considered a 2D image. Hence, different DL techniques will be looked into to estimate the channel. The main aim were as follow:

- **Proposing an efficient DL technique for channel estimation**
- **Determining the appropriate number of pilot signals fo restoration.**

Chapter 2

Wireless Communication

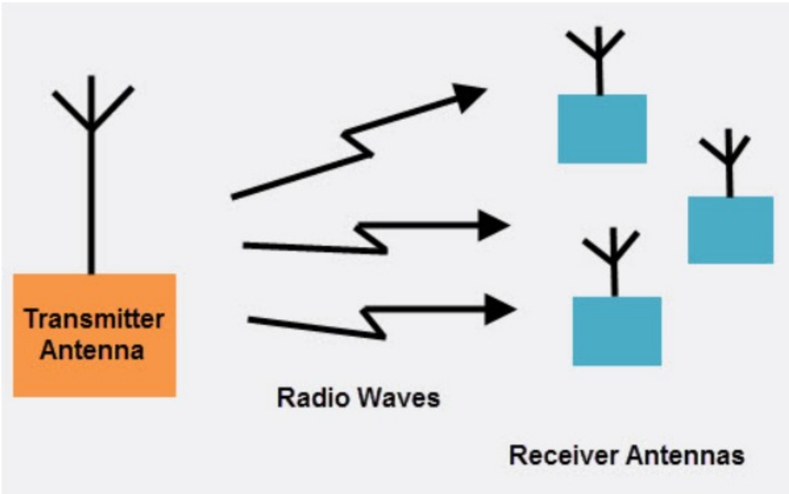

Figure 2.1: A typical wireless communication system

Communication Systems can be divided into two categories:

1. Wired Communication Systems

2. Wireless Communication Systems

On the basis of the use of cables or wires, and the choice of a medium. In wired communications, the medium can be a guided path through a cable which establishes a direct connection between two points.

On the other hand, in wireless communications, the messages are generally broadcasted into space and the medium is considered to be unguided due to the absence of single point to point communication, like in the case of wires. Instead of wires, antennas are used at the transmitter and receiver for transmission of EM waves.

Our analysis consists of estimating the channel parameters for wireless communication so that their effect can be taken into account and can be eliminated at the output to get the correct transmitted signal.

Figure 2.2: General Wireless Router Network

Wireless communication involves the transfer of information from the transmitter (Tx) to the receiver (Rx) without the use of connecting wires or a medium. The range for wireless communication systems varies from very short ranges, for example, a bluetooth speaker connected to the laptop, to very high ranges, including communications between two satellites.

2.1 Constituents of Wireless Networks

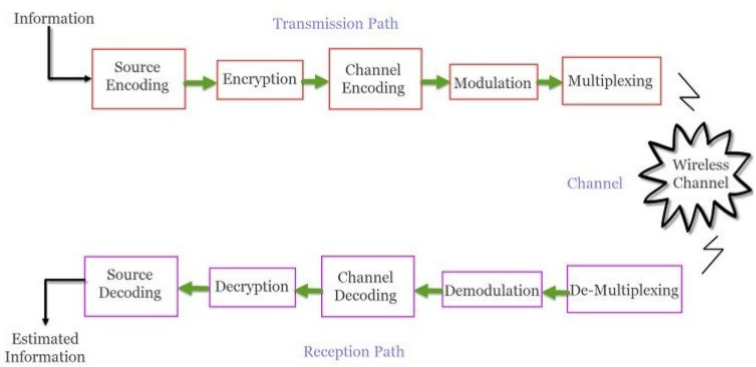

Figure 2.3: Constituents of a wireless network

Wireless Communication Systems can be divided into three main parts:

1. Transmitter
2. Channel
3. Receiver

Our analysis is mainly focused on the channel characteristics of the system and compensating for the channel at the receiver. This component is highly susceptible to noise and interference in the network and can lead to low SNRs(Signal to Noise Ratio) and high BER(Bit Error Rates).

2.2 Wireless over Wired Communication

Wired communications were predominantly used till the 20th century. However the advent of wireless communication technology resulted in a number of benefits, some of which are:

1. Economically effective
2. Mobile components
3. Reliability
4. Flexibility and Convenience
5. Easy installation

Although wireless communications come with a lot of benefits, it also has some shortcomings. The data transmission is done in space, which poses a problem of interception of the signal, or man-in-middle threats. Apart from this, one of the main drawbacks of broadcast communication is that it is prone to experiencing a lot of interference from multiple sources, when compared to wired communications. This includes interference from other wireless networks, in addition to environmental factors such as heavy rainfall, cloudy weather, etc. These have a quintessential standing in determining the channel characteristics of the network.

2.3 Relevance in 5G communications

Channel Estimation has a wide range of applications in hot topics like 5G wireless communications. 5G communications have extremely high speeds and reliability and use mmWaves for transfer of information. These are extremely high frequency waves which have undergone high attenuations due to environmental factors such as weather, humidity, rainfall and possible natural calamities like hurricanes. mmWaves also require line of sight transmission and obstructions in the channel can drastically hinder propagation. This is one of the main reasons why channel estimation becomes even more important in mmWave communications and in-turn 5G networks.

Another important aspect of 5G communications is that they have a large number of transmitters and receivers. These massive MIMO(multiple input multiple output) systems comprise a huge number of transmitter-receiver antennas in order to improve the speed and reliability of the network. However, this poses a challenge of a large overhead, which in-turn results in large dimensions of the channel matrix. Prediction of the channel characteristics can be extremely helpful in this case as it significantly reduces the overhead, and motivates us to find effective techniques with high accuracy and low complexity.

Chapter 3

Channel Estimation

The introduction of multiple input multiple output antennas in the system poses an increase in capacity. This is because of the increased diversity of the space (medium) between different transmitter and receiver antennas. Our job is to get an estimate of the channel and compensate for it at the receiver. The channel matrix is a two dimensional structure divided on the basis of frequency and time. In a particular time slot, signals at different frequencies are sent over the channel and their outputs are recorded at the receiver. Our approach would include estimation using a generalized model, using various techniques. For getting a better understanding as to how we approach channel estimation, we need to establish the difference between Data-aided approaches and Blind approaches.

3.1 Pilot Signals

Pilot signals are reference signals which are transferred from the transmitter to the receiver, and their ideal output is known at both the ends. These are single tone signals, that is, they have a particular frequency, and the frequency and the time slot in which they are transmitted are known, and can be used to compensate for the channel to know the actual transmitted value.

Pilot signals are transmitted in various fashions alongside the data. This information is also known to both the transmitters and the receivers. There are three main types of pilot arrangements:

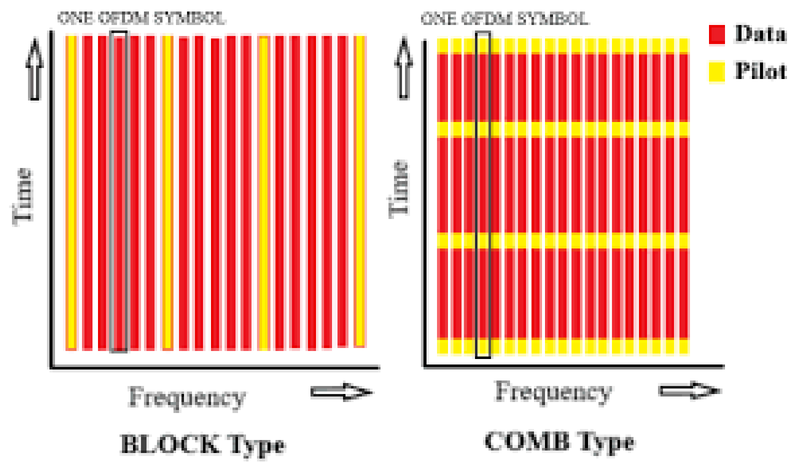

Figure 3.1: Block Types in an OFDM system

In our approach, we particularly choose the lattice type arrangement because it gives information of pilots over all frequencies within different time slots.

3.2 DATA AIDED APPROACH VS BLIND APPROACH

Different approaches for channel estimation are based on the extent to which the distribution of data is known. In the data-aided estimation, the approach relies on knowing the distribution of the data entirely, at both the ends of the communication network, that is, the transmitter as well as the receiver. This approach can be based on the known outputs of the pilot signals sent from the receiver, which can help in training our model in order to estimate the channel.

In the blind estimation, we are not aware of any statistics that the channel model follows, and we just have the received data to predict the channel characteristics. This helps in reducing the data overhead to transfer channel statistics but, however, has a higher chance of leading to a less accurate model. As is evident from the approaches, the data aided model has a higher accurate model prediction, but it has a higher data overhead. Thus, the data-aided approach requires a relatively higher bandwidth.

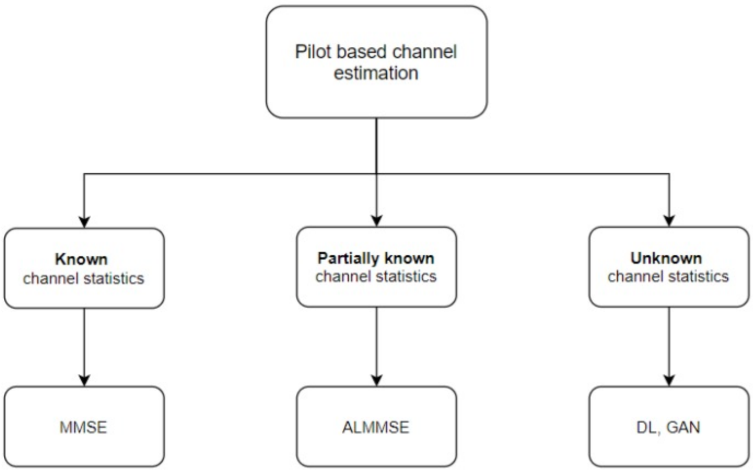

Figure 3.2: Channel statistics knowledge based classification

In our approach, we try to predict the channel model using different numbers of pilot signals, and plot them with respect to the accuracy achieved on the model, thus satisfying the trade off between accuracy and overhead. We try to use deep learning techniques with the blend of known pilot signals, in order to achieve maximum accuracy.

Chapter 4

Channel Estimation Methods

This chapter discusses some commonly used techniques for channel estimation along with the latest advancements in many of them as well. We will also show why the need for better methods arise as we go through this chapter.

4.1 Least Square (LS)

The Least Square estimator minimises the distance between the square of received signal and the actual signal. This goes by the name of the minimum variance unbiased estimator as well. In the approach, the channel and the noise distributions are both unknown. This estimator estimates the channel at pilot positions only and uses linear interpolation to fill up the entire channel matrix. It's channel estimate is given by,

$$H_{LS-estimate} = YP^H(PP^H)^{-1}, \tag{4.1}$$

where **Y** is the signal received matrix and **P** is pilot signals matrix.

Though the approach is less complex and doesn't require priori information, it doesn't provide a high estimate accuracy. Therefore, MMSE is a better estimator than LS.

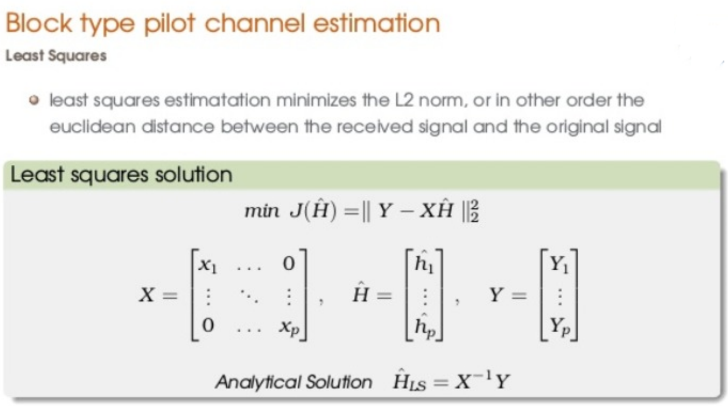

Figure 4.1: LS block diagram

4.2 Minimum Mean Square Error (MMSE)

MMSE or the Minimum Mean Square Error estimator minimises the mean square error of a dependent variable. It takes into account the channel statistics, namely the channel covariance matrix and the noise distribution which constitutes as the priori information to define the likelihood function in a much better way for minimising the error of reconstruction efficiently. Bayesian estimate is its other popular reference. It models the systems as follows:

$$vec(H) \sim \mathcal{CN}(0,\, R), \qquad vec(N) \sim \mathcal{CN}(0,\, S) \tag{4.2}$$

The the final estimate of the channel matrix as given by the estimator is,

$$vec(H_{LS-estimate}) = (R^{-1}) + (P^T \otimes I)^H S^{-1} (P^T \otimes I))^{-1} (P^T \otimes I)^H S^{-1} vec(Y) \tag{4.3}$$

The benefit of using the MMSE estimator is that it reduces the training sequence and provides a benchmarking estimate but without the channel statistics known beforehand, the quality of the results degrades and the process becomes too complex.

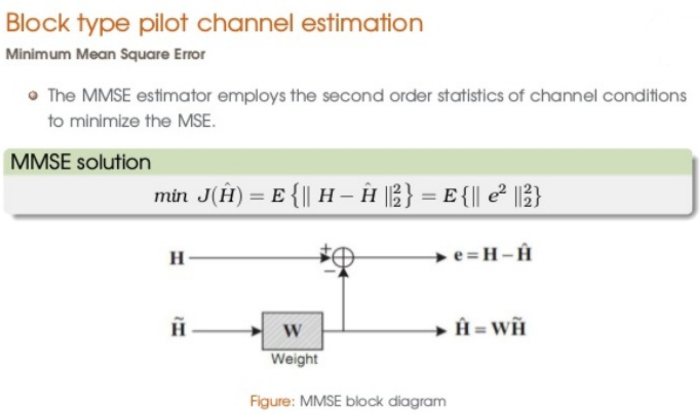

Figure 4.2: MMSE block diagram

4.3 Denoising Convolutional Neural Network

Denoising Convolutional Neural Network (DnCNN) in [1] is a deep learning technique that is used for the purpose of denoising. The network is capable of blind Gaussian denoising where the noise level is unknown and could be anything. A single network is capable of solving three general image denoising tasks better than the models which are trained to work for specific noise levels. The three tasks are namely blind SISR, JPEG deblocking and Gaussian denoising. Using the residual learning strategy and batch normalization , DnCNN in the hidden layers without the explicit knowledge, removes latent clean image. DnCNN prepared with a specific clamor gives preferred outcomes over best in class techniques, for example, BM3D, WNNM and TNRD. Also, DnCNN has faster training and boosted image denoising performance than the conventional CNN methods.

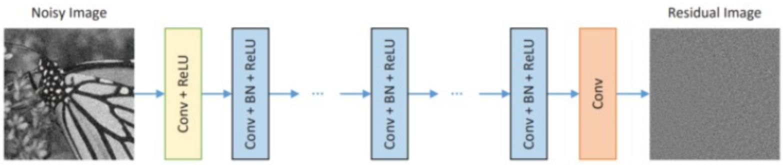

Figure 4.3: DnCNN Model Framework

Let's assume that there are D layers in the DnCNN model, further those layers are of 3 types:

1. **Conv+ReLu**: This layer has number of filters as 64 of the kernel size as 3x3×a, where a may vary depending upon image type.

2. **Conv+BN+ReLu**: These layers range from 2 to 18 and have batch normalization added in-between the convolutional layer and the ReLu unit. The size of the filters are 3x3x64 with 64 feature maps each.

3. **Conv**: The last layer is a convolutional layer only with filter size as 3x3x64.

4.4 Super Resolution Convolutional Neural Network

SrCNN or the Super Resolution Convolutional Neural Network in [2] is typically used for super resolution of image which is a standard problem. This is not a complex model as any other typical CNN model. It consists of 3 parts namely , patch extraction and representation, non-linear mapping, and reconstruction. It uses very few numbers of layers and filters yet achieves a very fast speed on even CPU because of its fully feed forward networks and no need of any optimisation. Also, the results of the model could be further improved by increasing the size of the dataset and size of the filters to build a deeper network.

1. **Patch extraction and representation**: Bicubic interpolation is first use to upscale the low resolution image to desired size before feeding the image to the SrCNN network. Patches of image that are overlapping patches from the low resolution image are extracted and considered as HR feature maps.

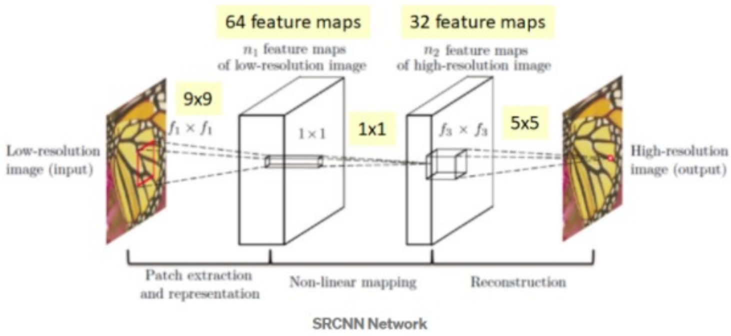

Figure 4.4: SrCNN Model Framework

2. **Non-linear Mapping:** A non-linear mapping from LR vector to a HR vector is performed much like in PCA fashion but in a nonlinear sense. Another feature map set is created from these vectors.

3. **Reconstruction:** Convolution is performed once again to reconstruct the image by aggregating the high resolution image patch representations.

Chapter 5

Model Framework

5.1 Channel Structure

We have taken a single pair of transmitter link and receiver link, i.e. a system that is SISO as in [4]. The channel consists of multiple sub carriers which transmit signals after a particle interval. Therefore, the channel matrix is thought of as a 2D matrix with the rows as the sub carriers and the columns as the time slots. Since, we are using deep learning methods, H or the channel time-frequency response matrix (of size $N_S \times N_D$) is considered as separate 2 Dimensional images for the imaginary values and real values respectively since the channel matrix might have complex values.

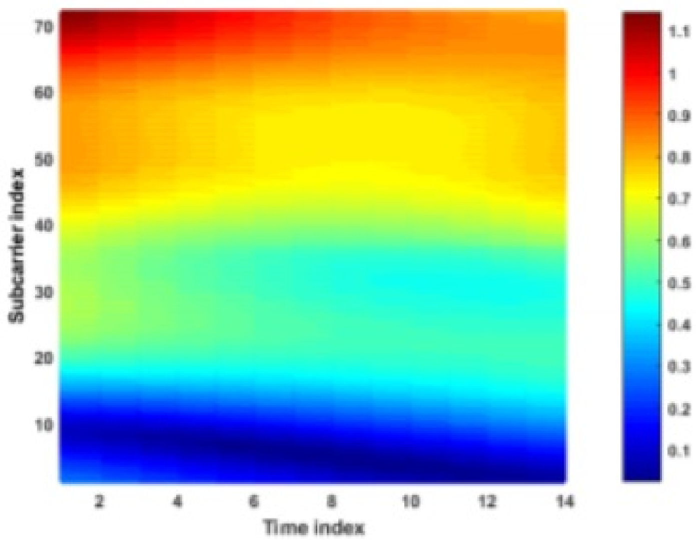

Figure 5.1: Time-frequency channel matrix as a normalized 2D image

5.2 Model Structure

An end to end pipeline is created to reconstruct the channel matrix from the pilot signals. The framework uses a two step process to reconstruct the channel, by first obtaining the values of the channel at the pilot position, then using a linear interpolation followed by noise removal to generate the matrix completely. More formally, the framework does two steps:

- A SR network takes the estimated values of the pilot positions of the channel H, an LR image, then estimates the unknown value of H to form a HR image.
- An IR network, which is a denoising network, filters out the noise from the previously estimated H to successfully reconstruct the actual channel.

The SR network is a linear model, whereas the IR network is a DnCNN model. The linear model increases the resolution of the image but is limited in quality due to its linear nature. DnCNN is a deep learning network which has around 20 layers. This network is efficient in removing the noise from the SR image so that the estimate is similar to the original channel.

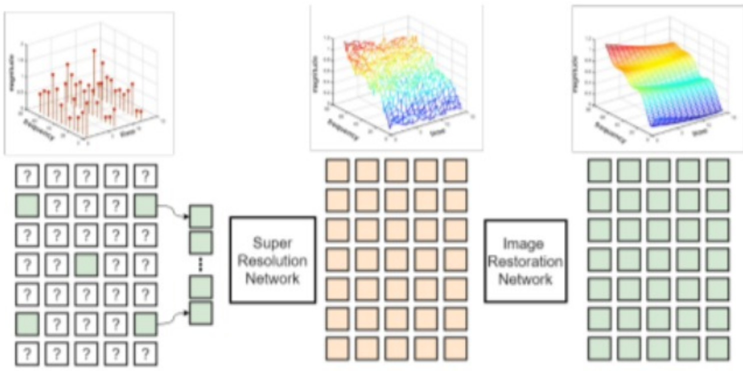

Figure 5.2: General pipeline for channel reconstruction

5.3 Training

The loss function is mean squared error or the mse. This error is basically the difference in the received signal data from train and test values and is represented by,

$$C = \frac{1}{\|\mathcal{T}\|} \sum_{\mathbf{h}_p \in \mathcal{T}} \|f(\Theta; \hat{\mathbf{h}}_p^{\mathrm{LS}}) - \mathbf{H}\|_2^2,$$

where τ is the training dataset, $\mathbf{H}$ is the channel matrix that is accurate and $\hat{\mathbf{h}}_p^{LS}$, pilot values vector. Since two steps are involved, the loss function is distributed between them. The minimisation of the Super Resolution network is done as follows, C_1,

where $\mathbf{Z} = f_S(\Theta_S; \hat{h}_p^{LS})$ is the product of the first network that is employed for SR. Next, the minimisation of the loss of network that is employed for IR C_2,

$$C_1 = \frac{1}{\|\mathcal{T}\|} \sum_{\mathbf{h}_p \in \mathcal{T}} \|\mathbf{Z} - \mathbf{H}\|_2^2,$$

$$C_2 = \frac{1}{\|\mathcal{T}\|} \sum_{\mathbf{h}_p \in \mathcal{T}} \|\hat{\mathbf{H}} - \mathbf{H}\|_2^2,$$

where $\hat{H} = f_R(Z; \Theta_D)$.

The dataset is generated artificially, and no modulation/demodulation is taken into account while training.

Chapter 6

Simulation Results

The data is artificially generated for testing the channel estimation technique. An ideal channel matrix is generated and some noise is added to it. The produced data is 2D data with the dimensions as 72x14, where 72 is the number of sub-carriers and 14 is the number of time slots. Below is the snippet for loading the dataset.

```
# load datasets
channel_model = "VehA"
SNR = 22
# perfect = loadmat("Perfect_" + channel_model + '.mat')['My_perfect_H']
perfect = loadmat("../input/btp-channel-estimation/Perfect_VehA.mat")['My_perfect_H']

# noisy_input = loadmat("Noisy_" + channel_model + "_" + "SNR_" + str(SNR) + ".mat")["My" + "_noisy_" + "H"]
noisy_input = loadmat("../input/btp-channel-estimation/Noisy_VehA_SNR_22.mat")["My" + "_noisy_" + "H"]
```

Figure 6.1: Loading Dataset

Two techniques have been implemented for channel estimation. For the purpose of Super Resolution (SR), we consider two models i.e. Linear SR Network and SrCNN Network. For Image Denoising, we use the standard DnCNN network.

6.1 Simulation of Super Resolution Network

The sections below discuss about both the approaches for super resolution along with the relevant code snippets used for their implementation.

6.1.1 Linear Super Resolution

The standard libraries such as keras and tensorflow have been used because of their integration with GPU that results in faster simulation. The batch size is taken as 128, and the learning rate $\alpha = 0.001$. The number of iterations don't exceed 500. Below is the code snippet for the linear model. Below is the snippet for the discussed model.

```
def SRCNN_model():
    input_shape = (72,14,1)
    x = Input(shape = input_shape)
    c3 = Convolution2D( 1 , (5 , 5) , kernel_initializer = 'he_normal', padding='same')(x)
    #c4 = Input(shape = input_shape)(c3)
    model = Model(inputs = x, outputs = c3)
    ##compile
    adam = Adam(lr=0.001, beta_1=0.9, beta_2=0.999, epsilon=1e-8)
    model.compile(optimizer=adam, loss='mean_squared_error', metrics=['mean_squared_error'])
    return model
```

Figure 6.2: Linear SR Model

6.1.2 Non-Linear Super Resolution

The standard libraries such as keras and tensorflow have been used because of their integration with GPU that results in faster simulation. The batch size is taken as 128, and the learning rate $\alpha = 0.001$. The number of iterations don't exceed 500. Below is the code snippet for the linear model. There are three convolutional layers with activation function chosen as *relu*. Below is the snippet for the discussed model.

```
def SRCNN_model():
    input_shape = (72,14,1)
    x = Input(shape = input_shape)
    c1 = Convolution2D( 64 , (9 , 9) , activation = 'relu', kernel_initializer = 'he_normal', padding='same')(x)
    c2 = Convolution2D( 32 , (1 , 1) , activation = 'relu', kernel_initializer = 'he_normal', padding='same')(c1)
    c3 = Convolution2D( 1 , (5 , 5) , kernel_initializer = 'he_normal', padding='same')(c2)
    #c4 = Input(shape = input_shape)(c3)
    model = Model(inputs = x, outputs = c3)
    ##compile
    adam = Adam(lr=0.001, beta_1=0.9, beta_2=0.999, epsilon=1e-8)
    model.compile(optimizer=adam, loss='mean_squared_error', metrics=['mean_squared_error'])
    return model
```

Figure 6.3: SrCNN Model

6.2 Simulation of Denoising Network

The standard libraries such as keras and tensorflow have been used because of their integration with GPU that results in faster simulation. The batch size is taken as 128, and the learning rate $\alpha = 0.001$. The number of iterations don't exceed 500. Below is the code snippet for the linear model. There are twenty convolutional layers with activation function chosen as *relu*. The first layer is *Conv+ReLu*, the next 18 layers are *Conv+BN+relu* and the last layer is just *Conv*. Below is the snippet for the discussed model.

```
def DNCNN_model():
    inpt = Input(shape=(None,None,1))
    # 1st layer, Conv+relu
    x = Conv2D(filters=64, kernel_size=(3,3), strides=(1,1), padding='same')(inpt)
    x = Activation('relu')(x)
    # 18 layers, Conv+BN+relu
    for i in range(18):
        x = Conv2D(filters=64, kernel_size=(3,3), strides=(1,1), padding='same')(x)
        x = BatchNormalization(axis=-1, epsilon=1e-3)(x)
        x = Activation('relu')(x)
    # last layer, Conv
    x = Conv2D(filters=1, kernel_size=(3,3), strides=(1,1), padding='same')(x)
    x = Subtract()([inpt, x])   # input - noise
    model = Model(inputs=inpt, outputs=x)
    adam = Adam(lr=0.001, beta_1=0.9, beta_2=0.999, epsilon=1e-8)
    model.compile(optimizer=adam, loss='mean_squared_error', metrics=['mean_squared_error'])
    return model
```

Figure 6.4: DnCNN Model

6.3 Results

The efficiency of the two approaches, both the Linear Super resolution model and the Non-Linear Super resolution model cascaded with the Denoising CNN framework is shown below for 30 epochs for the SR model and 20 epochs for the DnCNN model respectively.

6.3.1 Linear Super Resolution Framework

The following plots correspond to the error vs the number of epochs for the linear SR network. This ensures that the model error reaches a minima on saturation, and is also a measure of the correctness of the model.

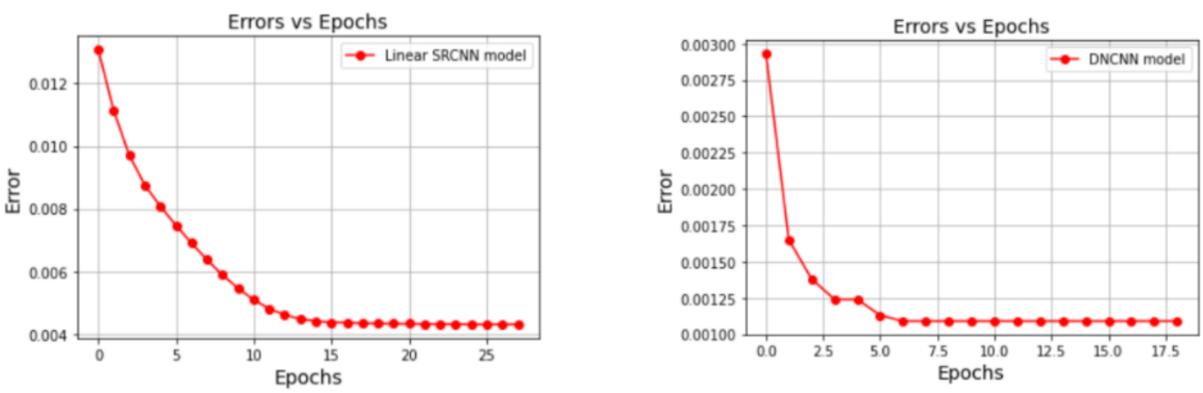

Figure 6.5: Error vs Epochs plots for Linear SR and DnCNN

6.3.2 Non-Linear Super Resolution Framework

The mentioned plots depict the error vs epochs for the non linear Super resolution model cascaded with the DnCNN model. Again, the efficiency keeps increasing as we increase the number of iterations on the training data, and attains a value on saturation after some time.

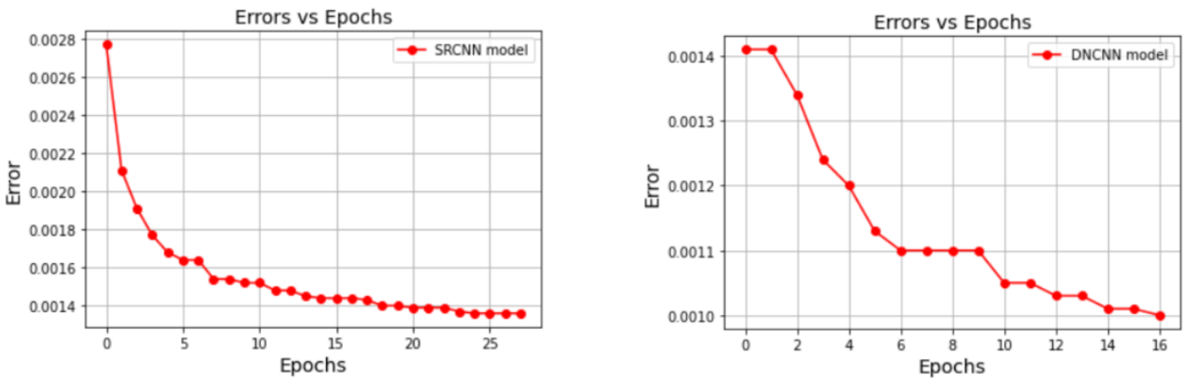

Figure 6.6: Error vs Epochs plots for Non-Linear SR and DnCNN

6.3.3 Efficiency vs Pilot Signals

Now we compare the efficiency of our models with respect to the number of pilot signals used for estimation.

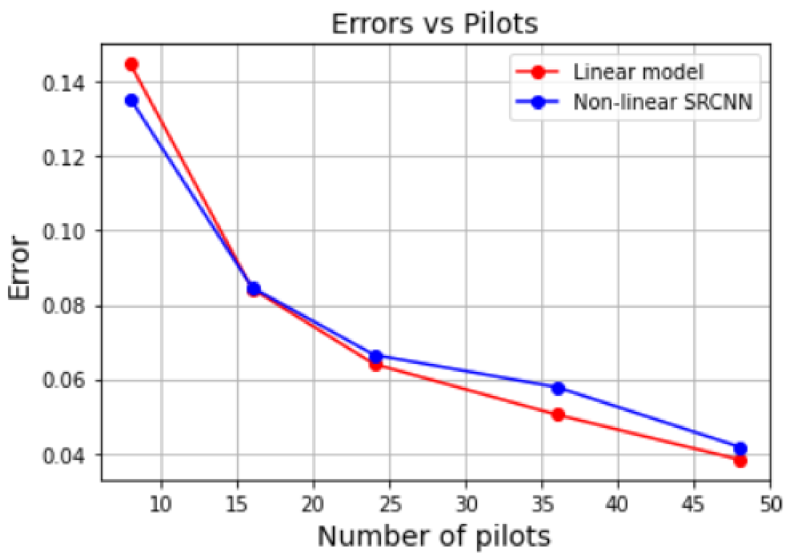

Figure 6.7: Error vs Number of Pilots for Linear and Non Linear SR

We obtain two conclusions from this plot:

1. The first conclusion is that the efficiency of the linear SR model and the non-linear SR model are comparable. Hence, we have achieved a significant complexity reduction after using the linear model, and the efficiency remains intact.

2. The mean squared error reduces with pilot signals, which implies that the efficiency increases as we use higher number of pilot signals. This occurs because of the fact that we have more amount of data to train our model, which in-turn is evident from the accuracy measure.

6.3.4 Complexity Analysis

We compare the complexity of both our approaches using several parameters, namely:

1. Time taken by the the models for training of both SR network and DnCNN networks, which has a direct correlation with the complexity of the models.

2. A comparison between the number of trainable parameters of both linear as well as non-linear SR models. Higher the number of parameters to be trained by the network, higher the time taken for training, and higher would be the complexity of the model.

Time Complexity Analysis

The following is the plot of the total time taken for training and prediction of the model with respect to the number of pilot signals.

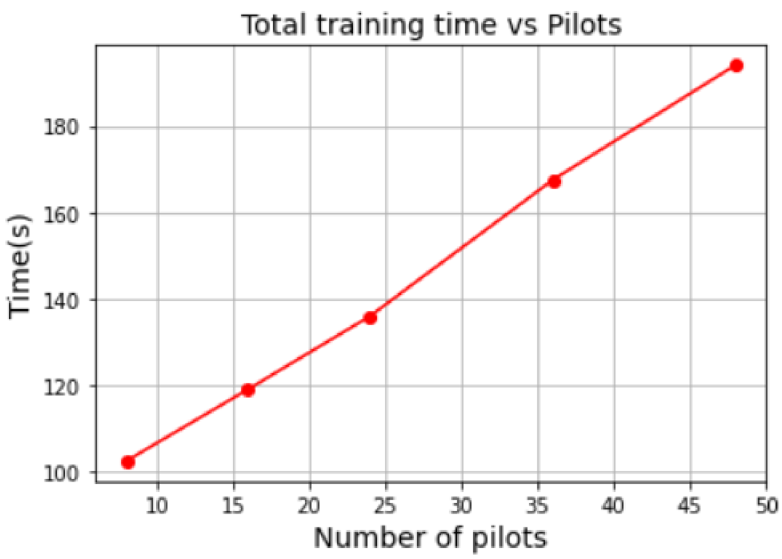

Figure 6.8: Time vs Number of Pilot signals

The plot is an increasing one which shows us that the time taken for the model to train is directly proportional to the number of pilot signals. This is in coherence with the theoretical analysis which depicts that increasing the number of pilot signals increases the amount of data that we have for training the model. Thus, the efficiency of the model increases as the number of pilot signals are increased, however, the complexity of the model also increases.

Now, we try to compare the training times for both our models, the linear SR and the non-linear SR cascaded with the DnCNN model as shown below.

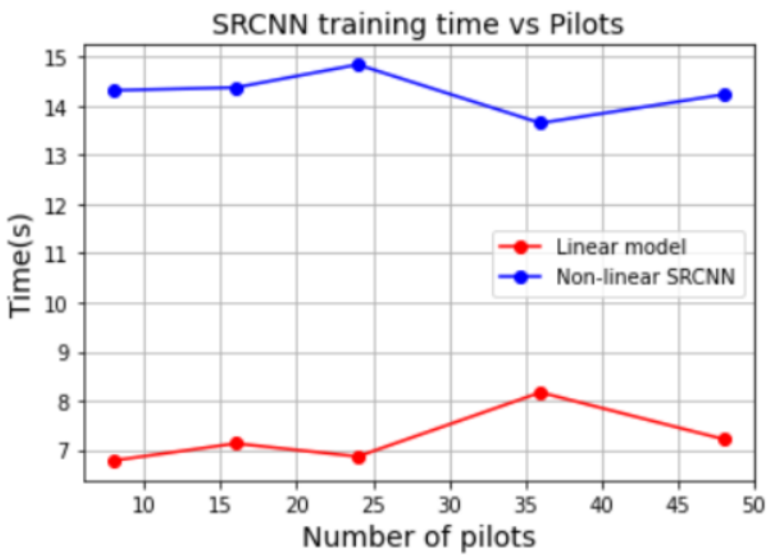

Figure 6.9: Comparison between time of training for both the models

The above figure depicts that the time taken to train the non linear super resolution model is more than double the time taken by the linear model that we introduced. There is more than 50% reduction in training time of the super resolution model. This shows that there is a significant reduction in complexity of the model, while having similar efficiency.

Trainable Parameters

The number of trainable parameters helps us in strengthening our claim that the linear model is considerably less complex than the first SrCNN model. The analysis is shown below in the bar graph:

The plot shows that the non linear model has 8129 parameters which need to be trained. When we use the linear model, this number reduces to 26, which is a significant decrease in complexity of the model.

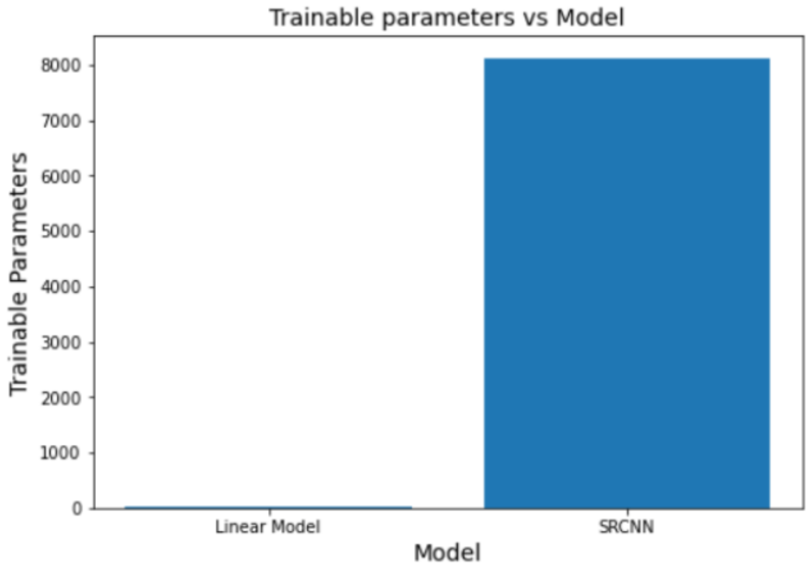

Figure 6.10: Trainable parameters for both the models

Finally, we conclude that the approach of using a linear super resolution model cascaded with a denoising neural network has improved complexity when compared to the non linear SrCNN cascaded with the denoising network. The complexity reduction is seen whilst having comparable accuracy of both the models.

Chapter 7

Discussion

7.1 Conclusion

In this project, we studied various techniques for channel estimation. We discussed the pros and cons of the conventional methods and the novelty of the deep learning techniques. Finally, we came up with a blend of a linear and a highly non linear model for estimating the channel. The frequency time response matrix of the channel was considered as 2 Dimensional images where the real and the imaginary parts were dealt with separately. Then the deep learning techniques such as super resolution and image denoising were exploited to reconstruct the channel only from a few pilot signals. The results were highly remarkable as the mean squared error was close to the Minimum Mean Square Error (MMSE) estimator which was considered to be the benchmark in our evaluation. We further showed that the time complexity was very less as compared to a typical dense DNN model. Hence, we triumphed on two fronts in terms of performance. The efficiency of the estimation was judged by the mean squared error between the actual channel and the reconstructed channel, and the time complexity was calculated using the time executed while training the model.

7.2 Future Work

Although the current model provides a reasonable accuracy in terms of error as well as time complexity, much work can still be done to improve the results. Since we were only able to improve the SR network and used the standard IR model much improvement in the second stage couldn't be done. The IR network could also be further explored as the current DnCNN uses a very large number of convolutional layers. Efforts could be made to make it a less complex model to reduce the time complexity. Hence, many avenues could still be explored.

Bibliography

[1] K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," in IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3142-3155, July 2017, doi: 10.1109/TIP.2017.2662206.

[2] C. Dong, C. C. Loy, K. He and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 2, pp. 295-307, 1 Feb. 2016, doi: 10.1109/TPAMI.2015.2439281.

[3] H. Ye, G. Y. Li and B. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," in IEEE Wireless Communications Letters, vol. 7, no. 1, pp. 114-117, Feb. 2018, doi: 10.1109/LWC.2017.2757490.

[4] Soltani, Mehran Mirzaei, Ali Pourahmadi, Vahid Sheikhzadeh, Hamid. (2018). Deep Learning-Based Channel Estimation.

[5] Sardar, Santu Mishra, Amit Khan, Mohammed. (2020). Indoor occupancy estimation using the LTE-CommSense system. International Journal of Remote Sensing. 41. 5609-5619. 10.1080/2150704X.2020.1734246.

[6] Umehara, Kensuke Ota, Junko Ishimaru, Naoki Ohno, Shunsuke Okamoto, Kentaro Suzuki, Takanori Ishida, Takayuki. (2017). Performance Evaluation of Super-Resolution Methods Using Deep-Learning and Sparse-Coding for Improving the Image Quality of Magnified Images in Chest Radiographs. Open Journal of Medical Imaging. 07. 100-111. 10.4236/ojmi.2017.73010.

[7] S. Colieri, M. Ergen, A. Puri and Bahai A, "A study of channel estimation in OFDM systems," Proceedings IEEE 56th Vehicular Technology Conference, Vancouver, BC, Canada, 2002, pp. 894-898 vol.2, doi: 10.1109/VETECF.2002.1040729.

[8] A. R. James, R. S. Benjamin, S. John, T. M. Joseph, V. Mathai and S. S. Pillai, "Channel estimation for OFDM systems," 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies, Thuckafay, 2011, pp. 587-591, doi: 10.1109/ICSCCN.2011.6024619.