IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 8, 2019, accepted July 19, 2019, date of publication July 25, 2019, date of current version August 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2931072

Path Loss Exponent and Shadowing Factor Prediction From Satellite Images Using Deep Learning

HASAN F. ATES, (Senior Member, IEEE), SYED MUHAMMAD HASHIR, TUNCER BAYKAS, (Senior Member, IEEE), AND BAHADIR K. GUNTURK
School of Engineering and Natural Sciences, Istanbul Medipol University, 34810 Istanbul, Turkey

Corresponding author: Bahadir K. Gunturk (bkgunturk@medipol.edu.tr)

This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under Grant 215E324.

ABSTRACT Optimal network planning for wireless communication systems requires the detailed knowledge of the channel parameters of the target coverage area. Channel parameters can be estimated through extensive measurements in the environment. Alternatively, ray tracing simulations can be done if the 3D model of the environment is available. One drawback of ray tracing simulations is the high computational complexity; therefore, ray tracing is not suitable for real-time coverage optimization. In this paper, we present a deep convolutional neural network-based approach to estimate channel parameters (specifically, path loss exponent and standard deviation of shadowing) directly from 2D satellite images. While deep learning methods require high computational resources for training and large amount of training data, once trained, the network can make predictions fast. Also, unlike the ray tracing simulations, there is no need for 3D model generation, and therefore, it can be applied easily using the images obtained from satellites or aerial vehicles. These make the proposed method a computationally efficient and reliable alternative to ray tracing simulations. The experimental results show that path loss exponent and large-scale shadowing factor at 900 MHz can be correctly classified by 88% and 76% accuracy, respectively.

INDEX TERMS Channel parameter estimation, path loss exponent, shadowing factor, deep learning.

I. INTRODUCTION

For wireless communication system operators, optimal network planning requires detailed knowledge of the channel parameters of the target coverage area. As long as the possible locations of the transmitters are low, channel parameter estimations can be obtained using results from extensive measurements in the environment [1], [2]. An alternative method is to employ ray tracing (RT) or ray launching (RL) simulations, which provide satisfactory performance compared to the real measurements [3], [4]. RT and RL simulations require 3D models of the environment, which could be obtained using computer vision techniques, such as stereo vision based depth estimation, applied on aerial images. 3D models of target areas are not always available, and it is not economic or practical to generate one for a target area in a short amount of time. Even if the 3D models are available, RT and RL simulations could be computationally costly and may not be suitable for real-time coverage optimization of large areas [5], [6].

Different models have been proposed to estimate path loss for terrestrial wireless networks. For example, the COST231 Walfisch-Ikegami (COST-WI) model is used for urban environments, and includes parameters such as average height of buildings, road widths, street orientation angles and building separation [7]. Such a detailed feature extraction is often not possible, and the performance of the model is not always satisfactory. In [8], a simplified model, including only the percentage of building occupation between the transmitter and receiver as the region feature, is presented. The building occupation percentage can be calculated from manually segmented images or through image segmentation directly applied to satellite images [9].

Machine learning techniques can also be used to predict path loss. For example, in [10], support vector machine (SVM) and artificial neural network (ANN) approaches, along with dimensionality reduction, are employed to

The associate editor coordinating the review of this manuscript and approving it for publication was Qinghua Guo.

predict path loss values. The feature vector includes height, thickness, and distance (from the transmitter) of the buildings between the transmitter and receiver. Other geometric features, such mean road width and mean building height, as well as transmitter/receiver heights and operation frequency have been considered in a number of papers based on ANNs [11]–[16]. For path loss prediction of regions other than urban areas, it is also possible to classify the region using aerial images as, for instance village or forest, and use the proper path loss model [17]. Recently, a review of existing machine learning based path loss prediction approaches, including support vector regression based [18] and decision tree based [19] methods, is presented [20].

The main disadvantage of most traditional machine learning approaches is that 3D models are required to make predictions. In our approach, the predictions are made directly from 2D images. We use 3D models only to create the dataset to be used for training the prediction network. Another disadvantage of existing methods is the need for hand-crafted features, which may not satisfactorily represent the actual characteristics affecting the prediction. In our approach, we use deep learning techniques, which extracts the relevant features through training. We will specifically utilize deep convolutional neural networks (CNNs), which incorporate convolutional layers [21], and have been successfully applied to many machine learning applications [22], [23].

In this paper, our goal is to model the relationships between channel parameters (specifically, path loss exponent and large-scale shadow factor) and 2D aerial/satellite images of an area. Supervised classification is used for predicting the range of parameter values in a given area. For this purpose, two different deep CNN architectures are trained and tested. Simulation results show up to 88% and 76% prediction accuracy for path loss exponent and large-scale shadowing factor parameters, respectively. The novel contributions of the paper can be summarized as follows:

- Deep learning is used to analyze and model the intricate dependencies between the channel parameters and 2D aerial/satellite images of an area. We show that channel parameters can be reliably predicted without the need for a 3D model or any auxiliary features of the region.
- Special deep learning techniques, such as transfer learning and multi-task learning, are successfully applied for accurate channel parameter prediction with limited training dataset.
- We propose a general framework which could be adapted to deep learning of other channel models and/or parameters at different communication settings. Deep learning provides a fast and cheap alternative to ray tracing simulations or extensive measurements in the field.

The paper is organized as follows. In Section II, we present the channel model and the prediction method. Deep learning techniques require large amount of data for training; in Section III, we explain the dataset generation process that we used. The network architectures and the parameter prediction performances are given in Section IV. We present an enhancement of the base prediction method in Section V. We discuss the results in Section VI and conclude the paper in Section VII.

II. CHANNEL PARAMETERS AND PREDICTION MODEL

The large-scale path loss model using a close-in free space reference distance is formulated as:

$$PL(d) = PL(d_0) + 10n \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma, \tag{1}$$

where $PL(d)$ is the path loss (in dB) at some distance d from the transmitter, $PL(d_0)$ is the path loss at a reference distance d_0, n is the path loss exponent and X_σ is a random variable with zero mean and standard deviation of σ, *i.e.*, the large-scale shadowing factor which refers to the amount of shadowing in the environment [24]. This model is widely used for estimating received signal strength in different wireless networks, including air-to-ground UAV communication [25].

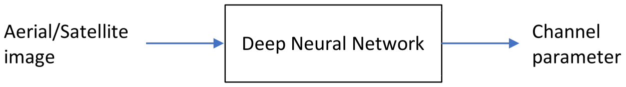


FIGURE 1. Channel parameter estimation. A 2D aerial/satellite image is input to a deep neural network to produce an estimate of the channel parameter for the corresponding region. For each channel parameter, a separate network is trained.

Our goal is to estimate the channel parameters (n and σ) of a region directly from a 2D satellite image of that region. As shown in Fig. 1, a deep neural network takes a satellite image as its input and returns a prediction of the channel parameter. In our case, there are two parameters (n and σ); so, for each parameter, there will be a separate neural network trained specifically for that parameter. The specific network architectures that we used will be given in Section IV.

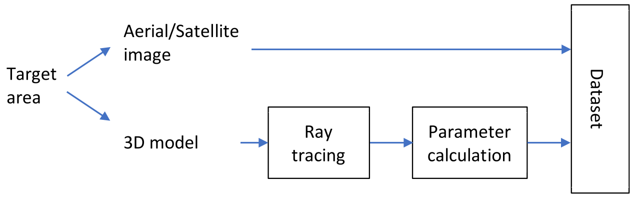


FIGURE 2. Dataset generation. A target region has both an aerial/satellite image and a 3D model. The 3D model is passed through ray tracing simulation to obtain path loss values, which are then used to calculate the channel parameters. The channel parameters and the corresponding 2D aerial/satellite image are added to the dataset.

III. DATASET GENERATION

Deep neural networks require large amount of training data. Fig. 2 shows the block diagram of the process to generate the dataset, which includes channel parameters and aerial/satellite images. For a target region, we have both the aerial/satellite image and the 3D model. The 3D model is passed through ray tracing simulation to obtain path loss values, which are then used to calculate the channel parameters.

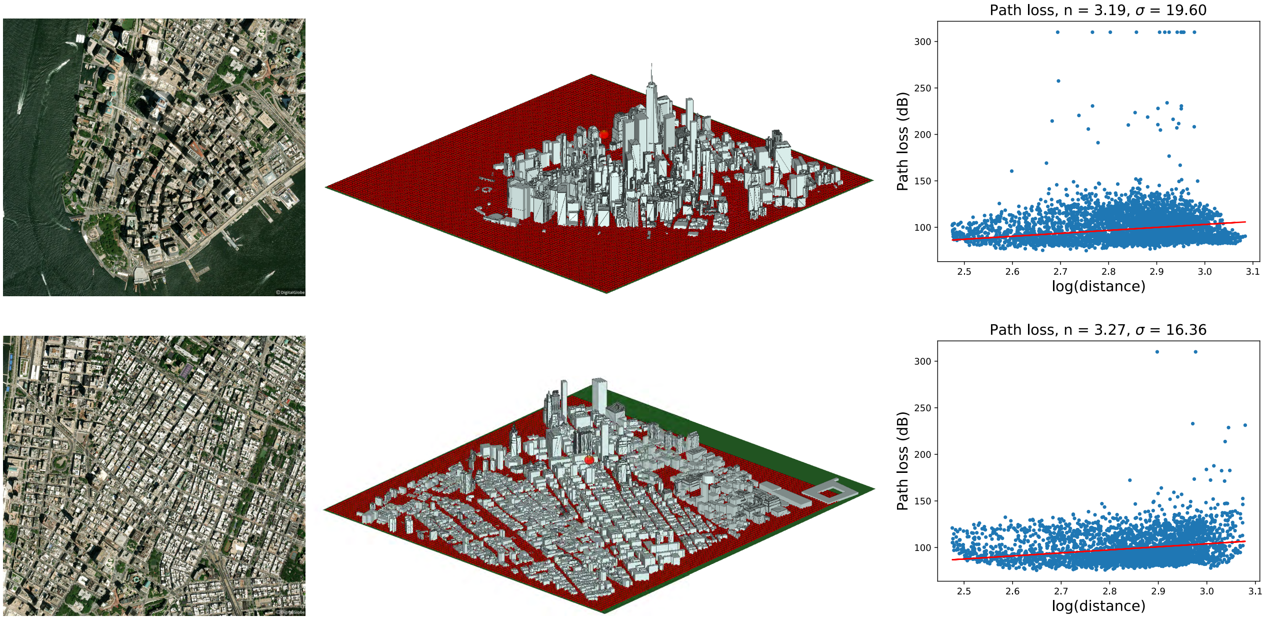


FIGURE 3. Satellite images, corresponding 3D models, and path loss measurements with predicted channel parameters are shown for two different regions.

The channel parameters and the corresponding image are added to the dataset. The process is repeated for all available regions with aerial/satellite images and 3D models to form the entire dataset.

Specifically, we used the SketchUp software[1] with the PlaceMaker plugin[2] to acquire the satellite images and the corresponding 3D models. Each image (and the corresponding 3D model) covers a 1.8 × 1.8 km area, mostly taken around the New York City. A 3D model is imported and merged with a flat terrain in Wireless InSite software.[3] Ray tracing simulation environment is created by placing a grid of 12,100 (110×110) receivers and a transmitter at the center of the 3D model. The receivers corresponding to positions inside the buildings are removed. The transmit power is set to 60 dBm and omni-directional antenna is chosen to achieve a good coverage area. The terrain is set as dry earth while the buildings are set as concrete in the simulation environment. The transmitter antenna is positioned at a height of 300 meters from the terrain to replicate the scenario of air-to-ground channel. The receivers are placed at 1.5 meters from ground level. (The simulation parameters are summarized in Table 1.) Ray tracing simulations are performed to generate the path loss values at each receiver. The path loss values are then used to calculate the wireless channel parameters (n and σ). Finally, the calculated channel parameters and the corresponding aerial image are added to the dataset.

At the end, a dataset consisting of a total of 999 images with channel parameters are generated using the process described above. In Fig. 3, satellite images, corresponding 3D models, and path loss measurements at the receivers (as a function of distance) along with the calculated channel parameters are shown for two different regions. Some other sample regions are shown later in the paper.

TABLE 1. Measurement setup parameters.

Transmission frequency	900 MHz
Transmit Power	+60 dBm
Antenna polarization	Vertical
Antenna radiation pattern	Omni-directional
Transmitter antenna height	300 m
Receiver antenna height	1.5 m
Transmitted signal	Sinusoid
Bandwidth	8 MHz
d_0	57.28 m
$PL(d_0)$	63.44 dB

IV. CHANNEL PARAMETER PREDICTION

We quantize the channel parameters (n, σ) into a certain number of sub-intervals; prediction of the correct (n, σ) sub-interval of each image is treated as a classification problem and solved by the use of deep learning architectures.

A. NETWORK ARCHITECTURES AND CLASS LABELING

There exist several successful deep CNN architectures used for classification in literature. In this paper, we test and compare two such architectures, namely VGG-16 [22] and ResNet-50 [23], for (n, σ) parameter classification. These networks (shown in Fig. 4) are well-known representatives of two alternative approaches. VGG-16 is a network with medium depth (16 layers) and more than 138M trainable

[1] https://www.sketchup.com/

[2] https://www.suplacemaker.com/

[3] https://www.remcom.com/wireless-insite-em-propagation-software

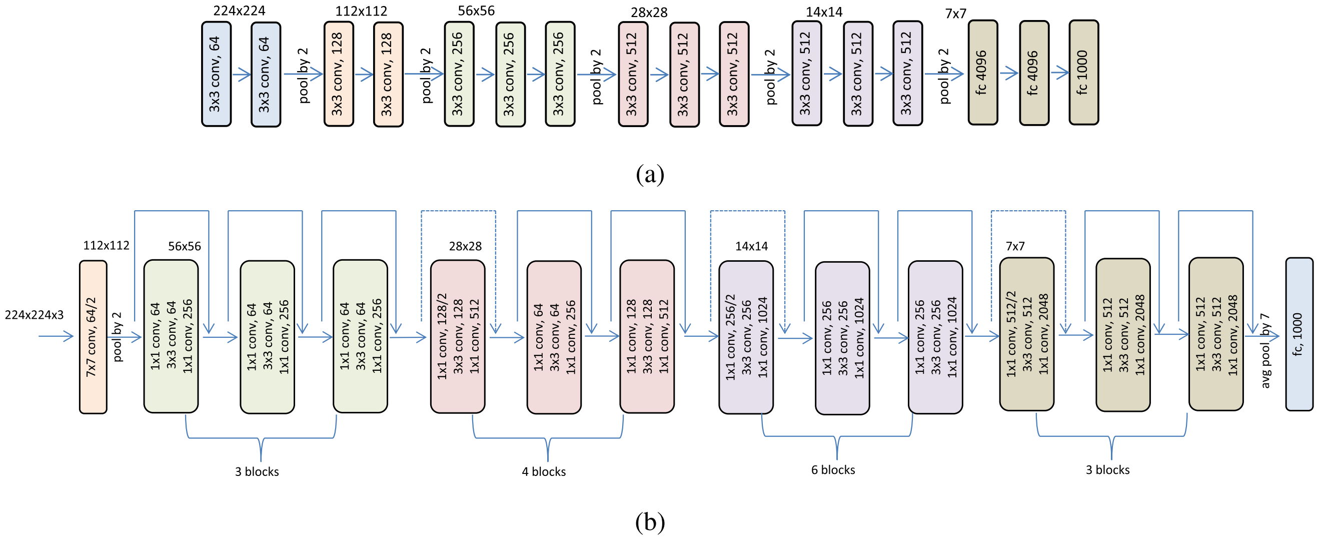


FIGURE 4. The architectures used in the experiments: (a) VGG-16 [22], (b) ResNet-50 [23]. In the illustrations, "conv" stands for convolution layer; "fc" stands for fully connected layer.

parameters, where most of the parameters are in the last three fully-connected layers. On the other hand, ResNet-50 is a deep (50-layer) fully convolutional network of about 25M parameters, where all layers are convolutional except for the final classification layer. We believe that it is important to analyze the impact of these architectural differences on (n, σ) classification performance.

In this paper, transfer learning is used to adapt and fine-tune pre-trained VGG-16 and ResNet-50 for the given problem. It is not easy to train a deep network from scratch, especially with a limited dataset as in our case. Transfer learning is the generally accepted strategy for the tasks where training set size is small. Both networks are pre-trained in the 1000-class ILSVRC ImageNet dataset [26]. The architectures shown in Fig. 4 are kept the same, except for the final classification layer. The number of outputs of the final layer is determined by the number of classes (i.e. sub-intervals) of (n, σ) parameters. The weights of this layer are trained and all other layers are fine-tuned using our training dataset.

We separate n and σ values into four and three sub-intervals, respectively, as indicated below. Such a quantization is sufficient for determining the correct communication model of a given region. The free-space path loss exponent n is equal to 2. For line-of-sight environments, especially in rural areas, average path loss is expected to be less than 2.5. In suburban and urban cellular environments, path loss coefficients between 2.5 to 3.5 are common, whereas in shadowed urban environments the path loss coefficient exceeds 3.5. Similarly, rural areas experience low shadowing factors $\sigma < 10$, and suburban and urban areas experience larger values. As a result, (n, σ) parameters are quantized as shown below and each interval is assigned a class label as follows:

$$l_n(n) = \begin{cases} 1, & n < 2.5 \\ 2, & 2.5 \leq n < 3.0 \\ 3, & 3.0 \leq n < 3.5 \\ 4, & 3.5 \leq n \end{cases} \tag{2}$$

$$l_\sigma(\sigma) = \begin{cases} 1, & \sigma < 10.0 \\ 2, & 10.0 \leq \sigma < 15.0 \\ 3, & 15.0 \leq \sigma \end{cases} \tag{3}$$

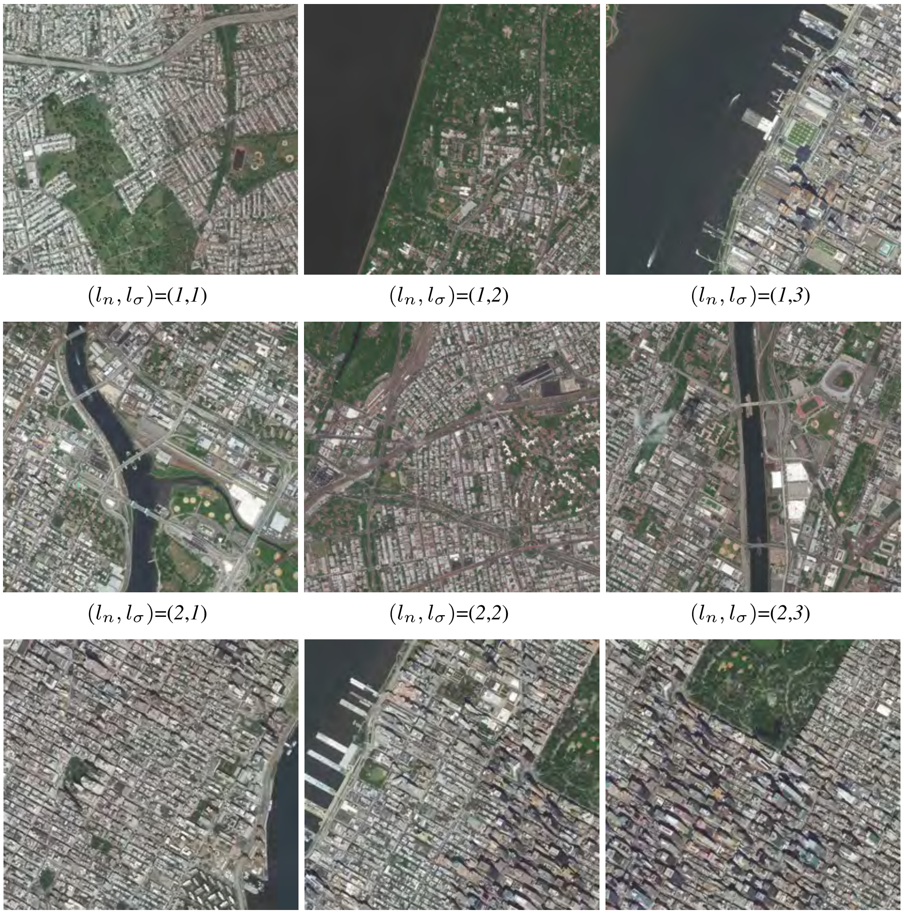
The distributions of the classes in our dataset is shown as a joint histogram of l_n and l_σ in Table 2. Fig. 5 provides sample images from nine different (l_n, l_σ) class combinations that are available in the dataset. As these figures suggest, both parameters n and σ increase as the density and height of buildings in the region increase. σ could be high when n is low (e.g. class $(l_n, l_\sigma) = (1, 3)$), if part of the region is densely covered with buildings but the rest is almost empty.

TABLE 2. Joint histogram for (l_n, l_σ).

		l_σ class		
		1	2	*3*
l_n class	*1*	193	95	10
	2	5	523	81
	3	0	29	40
	4	0	0	23

B. CLASSIFICATION PERFORMANCE

We perform 4-fold cross validation on the collected dataset. The dataset is divided into four random subsets; one subset is retained for testing and the remaining samples are used



(l_n, l_σ)=(1,1) (l_n, l_σ)=(1,2) (l_n, l_σ)=(1,3)

(l_n, l_σ)=(2,1) (l_n, l_σ)=(2,2) (l_n, l_σ)=(2,3)

(l_n, l_σ)=(3,2) (l_n, l_σ)=(3,3) (l_n, l_σ)=(4,3)

FIGURE 5. Sample images from (l_n, l_σ) classes.

as training data. This procedure is repeated for each subset and the overall prediction performance is reported on the whole dataset. Distinct VGG-16 and ResNet-50 architectures are trained and tested for classifying n and σ parameters. Since both architectures require fixed size inputs (see Fig. 4), the RGB satellite images are resized to $224 \times 224 \times 3$. The batch size is set to 16 for VGG-16 and 32 for ResNet-50. The learning rate is set to 0.001 for both architectures. We discuss the effect of these hyperparameters in Section VI.

SoftMax (SM) loss is used for the training of classification networks. The class scores at the network output are mapped to a probability distribution using softmax function and the negative log-likelihood of the true class is used as the training loss (L_{SM}):

$$L_{SM} = -\log(p_{i^*}), \quad \text{where} \quad p_i = \frac{e^{a_i}}{\sum_{k=1}^{N} e^{a_k}}, \tag{4}$$

N is the number of classes, i^* is the true class label, a_i is the ith class score, and p_i is the normalized probability of the ith class.

We evaluate the performance of proposed channel parameter prediction method by measuring the accuracy of predictions of the trained networks. Tables 3 and 4 provide the confusion matrices for VGG-16 and ResNet-50 architectures. The confusion matrices presented in Tables 3 and 4 show both overall and per-class accuracy for n and σ parameters respectively. Total accuracy gives the percentage of correctly

TABLE 3. Confusion matrices for l_n.

VGG-16		True class			
		1	2	3	4
Pred. class	1	242	38		
	2	56	561	25	1
	3		10	40	1
	4			4	21
Per-class acc.(%)		81	92	58	91
Total acc.(%)		86			

ResNet-50		True class			
		1	2	3	4
Pred. class	1	242	37		
	2	56	559	23	
	3		13	43	3
	4			3	20
Per-class acc.(%)		81	92	62	87
Total acc.(%)		86			

TABLE 4. Confusion matrices for I_σ.

VGG-16		True class		
		1	*2*	*3*
Pred. class	*1*	130	49	8
	2	63	542	92
	3	5	56	54
Per-class acc.(%)		66	84	35
Total acc.(%)		73		

ResNet-50		True class		
		1	*2*	*3*
Pred. class	*1*	123	37	8
	2	68	548	81
	3	7	62	65
Per-class acc.(%)		62	85	42
Total acc.(%)		74		

classified test images. Per-class accuracy is the percentage of correct labeling for each separate class of test images. For n, we get a total accuracy of 86%; and, for σ, we get a total accuracy of 73-74%. We discuss about the performance difference for n and σ in Section VI.

V. IMPROVING THE PERFORMANCE USING MULTI-TASK LEARNING

In the previous section, trying to learn only two parameters (n, σ) over the whole dataset causes problems of overfitting in the learning process. With a small-sized training set and with limited information per training sample, a deep network architecture could easily memorize the training data, causing loss of its generalization capability. Using transfer learning and just fine-tuning the pre-trained networks help alleviate the problem of overfitting. Yet, as further discussed in Section VI, we observe that, even though the training loss decreases continuously throughout the training, the total loss on the test set begins to increase after a certain number of iterations. The test accuracy, on the other hand, is more robust against overfitting; but it could reduce slightly after too many iterations.

Our dataset contains richer source of information than just the two channel parameters. The path loss is recorded for each receiver location on a regular rectangular grid. During training of the network, these path loss values could assist in learning the channel model parameters.

In this paper we do not consider point-wise estimation of path loss at every receiver location; instead, we propose to estimate the histogram of path loss throughout the entire region. We claim that histogram provides an informative summary of the levels of path loss (PL) in the region and it is an accurate indicator of how big n and σ should be. For this purpose, we compute 8-bin histogram h_{PL} of $(PL - PL_0)$ in the interval $[10, 90]$, where $PL_0 = PL(d_0) = 63.44$ dB.

The architectures of Section IV-A are modified by using an additional parallel fully-connected layer to learn h_{PL} as well as (n, σ). The additional layer has length-8 output to estimate the 8-bin histogram. The modified architecture is shown in Fig. 6. This approach is called multi-task learning [27] in literature and is successfully applied in jointly learning related tasks using a common architecture. In multi-task learning, the goal is to improve the learning performance by jointly learning the related tasks.

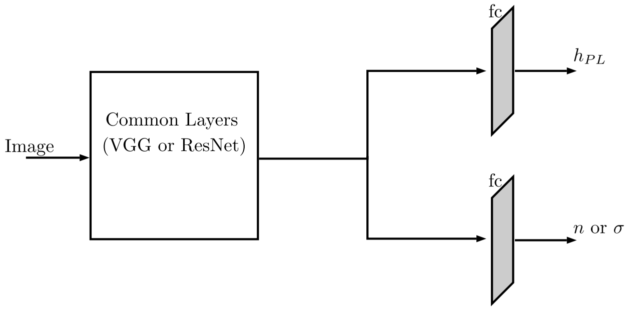


FIGURE 6. Multi-task learning architecture (fc: fully-connected layer).

While SoftMax loss function is used for learning (n, σ), Sigmoid Cross-Entropy (SCE) loss function is used for learning h_{PL}. The network outputs (b_i) are mapped to probabilities using sigmoid function and then the cross-entropy loss (L_{SCE}) is computed:

$$L_{SCE} = -\sum_{i=1}^{8} h_i \log(p_i), \quad \text{where} \quad p_i = \frac{1}{1+e^{-b_i}}, \tag{5}$$

and h_i are the true histogram values. The total loss is equal to the weighted sum of SM and SCE loss values:

$$TotalLoss = L_{SM} + \lambda L_{SCE}, \tag{6}$$

where λ is a parameter controlling the weights of the terms in the total loss function. (In the experiments, λ is set to 5; the choice of hyperparameters is discussed in the next section.)

TABLE 5. Confusion matrices for I_n.

VGG-16(MT)		True class			
		1	*2*	*3*	*4*
Pred. class	*1*	243	29		
	2	55	570	23	1
	3		10	44	1
	4			2	21
Per-class acc.(%)		82	94	64	91
Total acc.(%)		88			

ResNet-50(MT)		True class			
		1	*2*	*3*	*4*
Pred. class	*1*	246	34		
	2	52	566	22	
	3		9	45	1
	4			2	22
Per-class acc.(%)		83	93	65	96
Total acc.(%)		88			

The same set of simulations described in the previous section are repeated for deep architectures with multi-tasking (named as VGG-16(MT) and ResNet-50(MT)). Tables 5 and 6 provide the confusion matrices for multi-tasking. The confusion matrices presented in Tables 5 and 6

TABLE 6. Confusion matrices for l_σ.

VGG-16(MT)		True class		
		1	2	3
Pred. class	*1*	139	42	10
	2	54	551	79
	3	5	54	65
Per-class acc.(%)		70	85	42
Total acc.(%)		76		

ResNet-50(MT)		True class		
		1	2	3
Pred. class	*1*	140	39	6
	2	52	551	84
	3	6	57	64
Per-class acc.(%)		71	85	42
Total acc.(%)		76		

TABLE 7. Percentage accuracy for tested architectures.

Architecture	Accuracy for n (%)		Accuracy for σ (%)	
	Total	Av. per-class	Total	Av. per-class
VGG-16	86	81	73	62
ResNet-50	86	81	74	63
VGG-16(MT)	88	83	76	66
ResNet-50(MT)	88	84	76	66

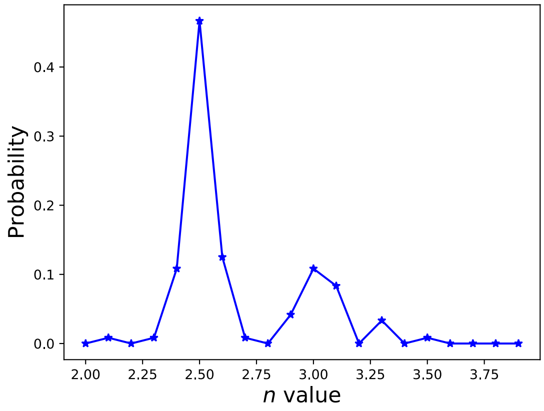


FIGURE 7. Histogram of *n* values for misclassifications.

show both overall and per-class accuracy for n and σ parameters respectively.

VI. DISCUSSION OF RESULTS

Table 7 summarizes the simulation results and provides total classification accuracy and mean per-class accuracy (i.e. average of per-class accuracies) for each tested architecture, with and without multi-tasking. Both network architectures exhibit similar performance in predicting l_n and l_σ. We see 2-4% improvement in accuracy values when using multi-task learning. While 88% total accuracy for l_n is highly successful, a maximum of 76% accuracy is achieved for l_σ.

Fig. 10 provides sample images with correctly classified n and σ parameters (for architecture ResNet-50(MT)). As noted before, regions with higher density of high-rise buildings correspond to classes with larger n and σ parameters. Fig. 11 provides images with misclassified n and σ parameters. Misclassified regions are mostly borderline cases where the building density in the region is in between the average densities of correct and false classes. For the images in Fig. 11(a),(b),(c),(d), the true values of n are 2.47, 2.53, 2.88 and 3.01, respectively, which are very close to the class boundaries of 2.5 and 3.0 between classes 1, 2 and 3. Fig. 7 shows the histogram of n values for the falsely labeled

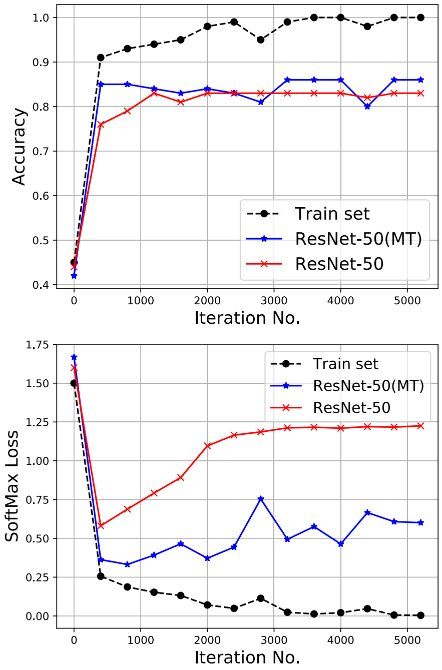


FIGURE 8. Convergence of loss and accuracy for *n*.

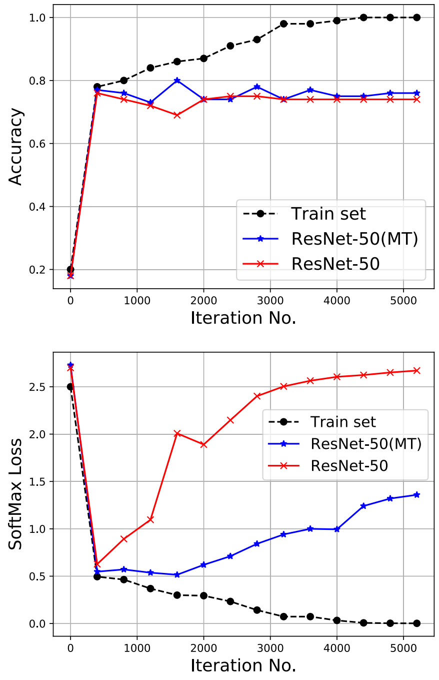


FIGURE 9. Convergence of loss and accuracy for σ.

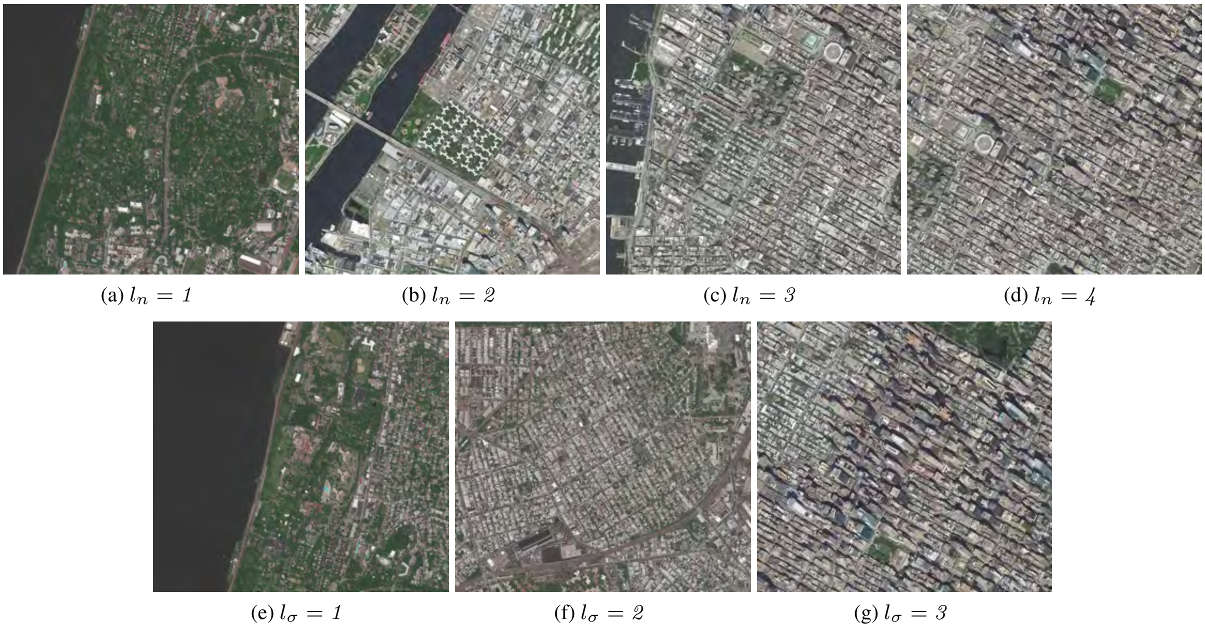


FIGURE 10. Correctly classified sample images for l_n and l_σ.

test images. In this figure, about 47% of all misclassifications are in the [2.45,2.55] range and 11% are in [2.95,3.05] range.

As seen from the confusion matrices, low prediction accuracy of $l_\sigma = 3$ class has a negative effect on the overall classification performance for l_σ. This class contains images with high variability in building density and height. The deep networks are capable of learning building density from 2D images, but not the building heights. The difficulty of inferring building height from 2D images is the main reason why prediction accuracy is lower for this class of regions.

The lack of building height information explains the performance difference in predicting n and σ parameters. While the value of n represents mean characteristics of a region, σ represents the deviation from the mean; and it is highly affected by shadowing due to high-rise buildings. For instance, the regions of 11(g) and (h) have similar n values of 2.84 and 2.89, despite quite different σ values of 12.68 and 16.59, respectively. An inspection of 3D model of region in Fig. 11(h) reveals mid-rise buildings towards the center where the transmitter is located. These buildings cannot be easily inferred from the 2D ortho-rectified view of the region.

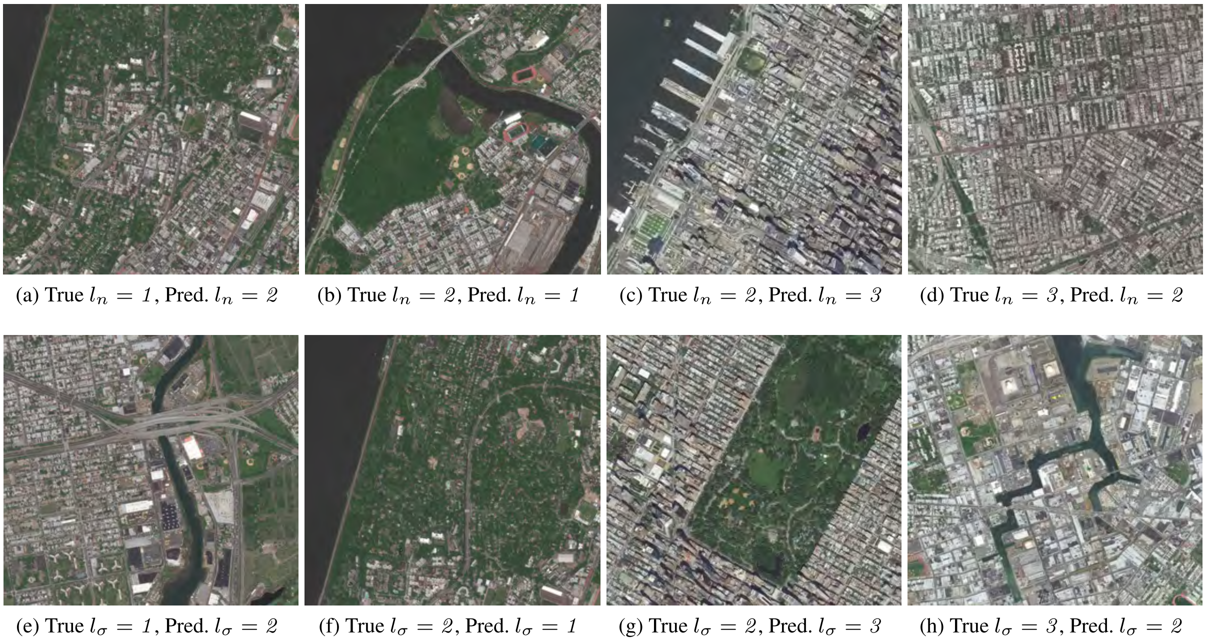
A. CONVERGENCE ANALYSIS OF LEARNING

As mentioned in Section V, training with a small dataset is prone to overfitting, which causes a loss of generalization capability of the network. Figs. 8 and 9 provide the change in test loss and accuracy vs. training iterations for n and σ respectively. Plots are provided for ResNet-50 and ResNet-50(MT). Training loss and accuracy for ResNet-50(MT) are also given as reference. We see that test accuracy reaches to its maximum in less than 1000 iterations and changes little afterwards. On the other hand, test loss starts to increase after making a dip at the early stages of training. The increasing loss is an indication of overfitting and it is more pronounced for ResNet-50 than for ResNet-50(MT). Test loss for ResNet-50(MT) increases as well, albeit at a lower pace. Since SoftMax loss is equal to the negative log-likelihood of true class, we can say that ResNet-50(MT) assigns on average higher likelihood to the true class label than ResNet-50. However, this improved likelihood results in only a couple of percent improvement in accuracy. With a larger training set, overfitting could be reduced and likelihood estimation could be improved. We can argue that, as the training set gets larger, it is more likely for ResNet-50(MT) to correct its mistakes and improve its accuracy than for ResNet-50.

B. ANALYSIS OF HYPERPARAMETERS

In this section, we discuss several hyperparameters of the learning procedure that might influence the classification performance. First of all, as noted above, two different architectures of VGG-16 and ResNet-50 have little difference in terms of classification accuracy. ResNet-50 could be preferable over VGG-16, since it has significantly lower number of weights, occupies less memory, is faster in training and inference despite being deeper. Interestingly, VGG-16 outputs lower test loss than ResNet-50 in most of the tested cases, implying that it suffers less overfitting despite having higher number of trained weights.

As for the hyperparameters of training, simulations show that prediction accuracy is robust under different settings. Different learning rates, batch sizes and loss weights for multi-tasking are tried. Increasing the batch size gives slight improvement in accuracy. We set batch size as 16 for



(a) True $l_n = 1$, Pred. $l_n = 2$
(b) True $l_n = 2$, Pred. $l_n = 1$
(c) True $l_n = 2$, Pred. $l_n = 3$
(d) True $l_n = 3$, Pred. $l_n = 2$
(e) True $l_\sigma = 1$, Pred. $l_\sigma = 2$
(f) True $l_\sigma = 2$, Pred. $l_\sigma = 1$
(g) True $l_\sigma = 2$, Pred. $l_\sigma = 3$
(h) True $l_\sigma = 3$, Pred. $l_\sigma = 2$

FIGURE 11. Misclassified sample images for l_n and l_σ.

VGG-16 and 32 for ResNet-50, by considering both prediction performance and GPU memory limitations. The weight (λ) of Sigmoid Cross-Entropy loss is tested in the range [3, 10] and little change in performance is observed. The effect of learning rate is also marginal.

It should also be noted that high resolution RGB satellite images are resized to 224 × 224 × 3 in order to be compatible with input dimensions of both VGG-16 and ResNet-50. We have also tested a modified VGG-16 architecture with input dimensions of 448 × 448 × 3 and have not observed any improvement in performance. Therefore, it is safe to say that downsizing does not cause a significant loss of features that could affect the prediction accuracy.

C. LIMITATIONS OF THE PROPOSED APPROACH

As mentioned before, the use of 2D RGB satellite images as input to the network makes it hard to learn building heights, which limits the prediction performance especially for σ parameter. In this paper, our goal is to develop a simple approach to channel parameter prediction from 2D satellite or aerial images, which could be used even when a 3D model is not available. As future work, the proposed architectures could be extended to include building height maps and/or digital elevation models as additional input to improve model prediction performance.

Large training set annotated with true channel parameters is necessary for supervised training of a deep network. In this paper, the dataset problem is handled by using Wireless InSite simulations based on available 3D models. Fine-tuning pretrained networks helps in learning a deep classifier with limited amount of training data. As future work, the dataset could be extended using different urban/suburban regions and cities. Also, actual channel simulations could be performed in the field to obtain real-life data.

Even though training a deep network requires high computational power and long training time, inference is actually fast when using optimized implementations of deep architectures. In our simulations, inference with VGG-16 and ResNet-50 take 20 ms and 16 ms per image, respectively, using *Caffe* framework on Tesla V100 GPU.

VII. CONCLUSION

In this paper, we presented a deep learning based approach to predict wireless channel parameters (specifically, path loss exponent and large-scale shadow factor) of an area directly from a satellite image of the area. The advantage of deep learning is that there is no need to explicitly define the features, which are learned as a part of the training process. We tested two different deep convolutional neural network architectures and presented an enhancement using multi-task learning, which incorporates the available path loss histogram into the training. We demonstrated the method on a specific scenario (frequency, transmitter/receiver heights); the method can be extended to other scenarios. For each scenario, the corresponding dataset should be generated and the network should be trained with that specific dataset. A further extension of this work could be clustering and analyzing the images according to their channel parameter values; this may lead to intuitive results about the region types and corresponding communication channel characteristics.

REFERENCES

[1] T. S. Rappaport, G. R. Maccartney, M. K. Samimi, and S. Sun, "Wideband millimeter-wave propagation measurements and channel models for future wireless communication system design," *IEEE Trans. Commun.*, vol. 63, no. 9, pp. 3029–3056, Sep. 2015.

[2] S. Sangodoyin, S. Niranjayan, and A. F. Molisch, "A measurement-based model for outdoor near-ground ultrawideband channels," *IEEE Trans. Antennas Propag.*, vol. 64, no. 2, pp. 740–751, Feb. 2016.

[3] B. Ai, K. Guan, R. He, J. Li, G. Li, D. He, Z. Zhong, and K. M. S. Huq, "On indoor millimeter wave massive MIMO channels: Measurement and simulation," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 7, pp. 1678–1690, Jul. 2017.

[4] M. Zhu, A. Singh, and F. Tufvesson, "Measurement based ray launching for analysis of outdoor propagation," in *Proc. 6th Eur. Conf. Antennas Propag. (EUCAP)*, Mar. 2012, pp. 3332–3336.

[5] F. A. Agelet, F. P. Fontan, and A. Formella, "Fast ray tracing for microcellular and indoor environments," *IEEE Trans. Magn.*, vol. 33, no. 2, pp. 1484–1487, Mar. 1997.

[6] G. Liang and H. L. Bertoni, "A new approach to 3-D ray tracing for propagation prediction in cities," *IEEE Trans. Antennas Propag.*, vol. 46, no. 6, pp. 853–863, Jun. 1998.

[7] D. J. Cichon and T. Kürner, "Propagation prediction models," Florence, Italy, Tech. Rep. COST-231 TD (95) 66, Apr. 1995, pp. 115–207.

[8] L. C. Fernandes and A. J. M. Soares, "Simplified characterization of the urban propagation environment for path loss calculation," *IEEE Antennas Wireless Propag. Lett.*, vol. 9, pp. 24–27, 2010.

[9] L. C. Fernandes and A. J. M. Soares, "On the use of image segmentation for propagation path loss prediction," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Oct. 2011, pp. 129–133.

[10] M. Piacentini and F. Rinaldi, "Path loss prediction in urban environment using learning machines and dimensionality reduction techniques," *Comput. Manage. Sci.*, vol. 8, no. 4, pp. 371–385, Nov. 2011.

[11] S. P. Sotiroudis, S. K. Goudos, K. A. Gotsis, K. Siakavara, and J. N. Sahalos, "Application of a composite differential evolution algorithm in optimal neural network design for propagation path-loss prediction in mobile communication systems," *IEEE Antennas Wireless Propag. Lett.*, vol. 12, pp. 364–367, 2013.

[12] S. P. Sotiroudis and K. Siakavara, "Mobile radio propagation path loss prediction using Artificial Neural Networks with optimal input information for urban environments," *AEU-Int. J. Electron. Commun.*, vol. 69, no. 10, pp. 1453–1463, Oct. 2015.

[13] I. Popescu, I. Nafornita, and P. Constantinou, "Comparison of neural network models for path loss prediction," in *Proc. IEEE Int. Conf. Wireless Mobile Comput., Netw. Commun.*, Aug. 2005, pp. 44–49.

[14] E. Ostlin, H.-J. Zepernick, and H. Suzuki, "Macrocell path-loss prediction using artificial neural networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 6, pp. 2735–2747, Jul. 2010.

[15] B. J. Cavalcanti, G. A. Cavalcante, L. M. D. Mendonça, G. M. Cantanhede, M. M. de Oliveira, and A. G. D'Assunção, "A hybrid path loss prediction model based on artificial neural networks using empirical models for LTE and LTE-A at 800 MHz and 2600 MHz," *J. Microw., Optoelectron. Electromagn. Appl.*, vol. 16, pp. 708–722, Sep. 2017.

[16] Y. Zhang, J. Wen, G. Yang, Z. He, and X. Luo, "Air-to-air path loss prediction based on machine learning methods in urban environments," *Wireless Commun. Mobile Comput.*, vol. 6, May 2018, Art. no. 8489326.

[17] M. Neunerdt, A. Engels, and R. Mathar, "Land use classification as a key component for path loss prediction in rural areas," in *Proc. 7th Int. Symp. Wireless Commun. Syst.*, Sep. 2010, pp. 666–670.

[18] M. Uccellari, F. Facchini, M. Sola, E. Sirignano, G. M. Vitetta, A. Barbieri, and S. Tondelli, "On the use of support vector machines for the prediction of propagation losses in smart metering systems," in *Proc. IEEE 26th Int. Workshop Mach. Learn. Signal Process.*, Sep. 2016, pp. 1–6.

[19] C. A. Oroza, Z. Zhang, T. Watteyne, and S. D. Glaser, "A machine-learning-based connectivity model for complex terrain large-scale low-power wireless deployments," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 576–584, Dec. 2017.

[20] Y. Zhang, J. Wen, G. Yang, Z. He, and J. Wang, "Path loss prediction based on machine learning: Principle, method, and data expansion," *Appl. Sci.*, vol. 9, p. 1908, May 2019.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014. *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[24] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[25] R. Amorim, H. Nguyen, P. Mogensen, I. Z. Kovács, J. Wigard, and T. B. Sørensen, "Radio channel modeling for UAV communication over cellular networks," *IEEE Wireless Commun. Lett.*, vol. 6, no. 4, pp. 514–519, Aug. 2017.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[27] Y. Zhang and Q. Yang, "A survey on multi-task learning," Jul. 2017, *arXiv:1707.08114*. [Online]. Available: https://arxiv.org/abs/1707.08114

HASAN F. ATES (S'96–M'04–SM'11) received the Ph.D. degree from the Department of Electrical Engineering, Princeton University, in 2004. He was a Research Associate with Sabanci University, from 2004 to 2005. He held positions of Assistant, Associate, and Full Professorship with Isik University, from 2005 to 2018. In 2018, he joined the Department of Computer Engineering, Istanbul Medipol University, where he is currently a Professor. He is the author or coauthor of more than 50 peer-reviewed publications in the areas of image/video processing/coding and computer vision.

SYED MUHAMMAD HASHIR received the B.E. degree in electronics engineering from the PAF-Karachi Institute of Economics and Technology, Pakistan. He is currently pursuing the M.S. degree in electrical engineering and cyber systems with Istanbul Medipol University, Turkey. His current research interests include wireless channel modeling, software-defined radios, wireless communications, and analog/RF circuit design.

TUNCER BAYKAS (S'01–M'07–SM'13) was an Expert Researcher with NICT, Japan, from 2007 to 2012. He is currently an Assistant Professor with Istanbul Medipol University. He has served as a Co-Editor and the Secretary for 802.15 TG3c and contributed to many standardization projects, including 802.22, 802.11af and 1900.7. He is also the Vice Chair of the 802.19 Wireless Coexistence Working Group and the 802.11 Task Group.

BAHADIR K. GUNTURK received the B.S. degree from Bilkent University, Turkey, and the Ph.D. degree in electrical engineering from the Georgia Institute of Technology, in 1999 and 2003, respectively. From 2003 to 2014, he was with the Department of Electrical and Computer Engineering, Louisiana State University. Since 2014, he has been with Istanbul Medipol University, where he is currently a Professor. His research interests include image processing and computer vision.