PATH LOSS PREDICTION FROM HEIGHT MAP USING DEEP LEARNING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF
ENGINEERING AND NATURAL SCIENCES
OF ISTANBUL MEDIPOL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL, ELECTRONICS ENGINEERING AND CYBER SYSTEMS

By
Mustafa Bal
February, 2021

Path Loss Prediction From Height Map Using Deep Learning
By Mustafa Bal
February, 2021

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Bahadır K. GÜNTÜRK (Advisor)

Asst. Prof. Dr. Muharrem MERCİMEK

Prof. Dr. Hasan F. ATEŞ

Approved by the Graduate School of Engineering and Natural Sciences:

Assoc. Prof. Dr. Yasemin Yüksel Durmaz
Director of the Graduate School of Engineering and Natural Sciences

Foreword

This thesis is written as completion of the master study in Electrical-Electronics Engineering and Cyber Systems program at Istanbul Medipol University.

The thesis focuses on computer vision and deep learning techniques such as regression and conditional gan-based approach from a height map to find the path loss of the targeted region.

Since the years that I have been conducting research on the topic. I have experienced this period as very interesting and instructive. In the beginning, I had little knowledge of legal writing and the concept of compensation. However, I have been able to achieve a result I am very satisfied with. I would like to thank my supervisor Prof. Dr. Bahadır K. Güntürk. Their valuable insights and directions gave me needful guidance to complete the research and write this thesis.

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: MUSTAFA BAL

Signature :

Acknowledgement

I would like to express my deep sincere gratitude to my advisor Prof. Dr. Bahadir K. Gunturk for all the support he generously gave to me during the research, and for his patience and motivation. Without his patience and commitment, this work would have never been completed. I learned a lot from him, and he was a great mentor to me.

I would like to give special thanks to Prof. Dr. Hasan F. Ates, Ass. Prof. Dr. Tuncer Baykas. My completion of this work could not have been accomplished without their support and guidance.

I thank my friends in Istanbul Medipol University for their help and fruitful discussions.

Finally, I would like to take this chance to mention that without the support of my family and loved ones, and without their encouragement and sincere love, nothing would have had a meaning at all.

Contents

List of Figures

List of Tables

List of Symbols

$FSPL$ Free-space path loss

PL Path loss

$G(h_{te})$ Transmitter altitude gain

$G(h_{re})$ Receiver altitude gain

d Distance

d_0 Reference distance

n Path loss exponent

σ Shadowing factor

x_n Input values

z_n Output values

w Angle of transmitter to receiver

θ Phase from receiver to transmitter

η Excessive path loss

dB Decibel

Abbreviations

ATG Air-To-Ground. 1, 2, 10–12, 27

cGAN Conditional Generative Adversarial Network. 3, 5, 16, 27, 30, 33

CNN Convolutional Neural Network. 14

DNN Deep Neural Network. 14

E-FS Electromagnetic Field Strength. 10

GAN Generative Adversarial Network. viii, 15, 16

GPU Graphical Processing Units. 12

KNN K-Nearest-Neighbor. 10, 11

LoS Line-Of-Sight. 28

MSE Mean Squared Error. 13, 25, 34

nLoS Non-Line-Of-Sight. 28

NN Neural Network. 14

RSRP Reference Signal Received Power. viii, 15

Rx Receiver. 1, 2, 6, 9, 11, 27

SGD Stochastic Gradient Descent. 25

SQ1 Sequential 1. ix, 30, 32

Tx Transmitter. 1, 11, 27, 42

UAV Unmanned-Aerial-Vehicle. 17, 18, 27, 49

WI Wireless InSite. 9, 10, 22

ÖZET

DERİN ÖĞRENME KULLANILARAK YÜKSEKLİK HARİTALARINDAN YOL KAYBI TAHMİNİ

Mustafa Bal
Elektrik-Elektronik Mühendisliği ve Siber Sistemler, Yüksek Lisans
Tez Danışmanı: Prof. Dr. Bahadır K. GÜNTÜRK
Şubat, 2021

Başarılı bir ağ planlamasının yapılabilmesi için hedeflenen kapsama alanının kanal parametreleri hakkında detaylı bilgiler gerektirmektedir. Işın izleme simülasyonlarıyla elde edilen kanal parametreleri yeterli bilgi vermektedir. Bu işlemde belirlenen alanın 3-boyutlu modeli simülasyona verilerek yol kaybı hesaplaması yapılmaktadır. Fakat ışın izleme simülasyonları yüksek hesaplama karmaşıklığı ve zaman gerektirmektedir. Bu tezde genellikle yol kaybı veya parametreleri tahmini için çeşitli derin öğrenme yöntemleri önerilmiştir. Yol kaybı veya parametrelerini tahmin etmek için iki çözüm vermekteyiz. İlk olarak, uydu ya da yükseklik haritası görüntüleri ile derin öğrenme yöntemleri kullanılarak yol kaybı kuvveti ve büyük ölçekli gölgeleme faktörü değerini tahmin etmek için regresyon modellemesi gösterilmiştir. Derin ağın eğitimi için gerekli veri kümesi ışın izleme simülasyonları ile üretilmiş, uydu ve ya yükseklik haritaları derin sinir ağına verilip, çıktı olarak da istenilen kanal parametrelerinin kestirimi regresyon yöntemiyle elde edilmiştir. Yol kaybı, kablosuz iletişimdeki parametrelerinin yanı sıra kritik bir değer olduğu için, ikinci sorunumuz koşullu genel düşmanca ağı kullanarak bir bölgenin noktasal aşırı yol kaybı değerlerini tahmin etmektir. Veri seti, çalışmalarımızın ilk probleminde olduğu gibi gereklidir, bu nedenle ışın izleme simülasyonları da bu sorunun gerçek değeri olarak üretilmiştir. Bu yöntemle, alıcının yol kaybı değerini her noktada doğrudan bulmayı hedeflenmektedir. Bu metodumuz noktasal tahmin için mükemmel bir model olmasa konvolüsyonlu ağlara nazara bölgeler için daha güvenilir bilgiler vermektedir. Elde edilen sonuçlar ayrıntılı olarak noktasal ve olasılık dağılımları olarak gösterilmiş ve analiz edilmiştir.

Anahtar sözcükler: Derin öğrenme, yükseklik haritaları, kanal parametrelerinin kestrimi, regresyon, yol kaybı kuvveti, gölgeleme faktörü, aşırı yol kaybı, insansız hava aracı, havadan karaya iletişim sistemi.

ABSTRACT

PATH LOSS PREDICTION FROM HEIGHT MAP USING DEEP LEARNING

Mustafa Bal
M.S. in Electrical, Electronics Engineering and Cyber Systems
Advisor: Prof. Dr. Bahadır K. GÜNTÜRK
February, 2021

Wireless channel parameters of a region are required for a successful network planning. Sufficient information about those parameters can be obtained by either actual measurements or ray-tracing simulations that use 3D model of the target area. However, measurements are costly and time consuming, and ray-tracing simulations have high computational cost. This thesis recommends various methods of estimating path loss or its parameters using deep learning. We give two solutions for estimating path loss or path loss parameters. Firstly, regression modeling is shown for estimating path loss exponent and shadowing factor of the wireless channel by using deep learning methods with satellite images or height map. Path loss dataset that is needed for training the deep neural network is produced by ray-tracing simulations. The deep network takes satellite image or height map as input and applies regression to estimate the desired channel parameters. Since the path loss is a critical value as well as its parameters in wireless communication, our second problem is to estimate the point-wise excessive path loss values of a region using the conditional general adversarial network. Ray-tracing simulations are also taken as the ground truth for this problem. With this method, we aim to find the path loss value of the receiver directly at each region. Even though it is not a perfect model for point-wise prediction but it can give us more reliable general information for the region than the convolutional networks. The results obtained are shown and analyzed as point-wisely for each region and probability distributions in detail.

Keywords: Deep learning, height maps, channel parameter estimation, regression, path loss factor, shadowing factor, excessive path loss, unmanned aerial vehicle, air-to-ground communication system.

Chapter 1

Introduction

Channel parameter estimation such as path loss and its parameters are crucial for wireless communication systems because wireless operators plan networks according to these parameters. Detailed and accurate channel parameter information is required to make the best network planning, such parameters are estimated with detailed measurements. The closest results to these measurements are the ray-tracing [2] simulations, but these simulations have high computational cost and therefore have ceased to be a very preferred method.

Besides to ray-tracing simulations, empirical models such as Okumura-Hata [10] and COST-231 Hata [11] can be used in path loss predictions and it can give different results depending on the type of area, such as urban, suburban, rural areas [12]. As observed in [13], modeling is sometimes required to obtain additional information such as building density between Transmitter (Tx) and Receiver (Rx). Even if the models are established correctly, the performance of these models does not always give satisfactory results.

Machine learning based approaches are used to predict path loss and delay spread [3] and the results show that empirical models produce much worse results for Air-To-Ground (ATG) wireless channel parameters than machine learning methods. Recently with the widely usage of computer vision methods [8, 14]

and deep convolutional architectures [15] that can use to obtain structural information from the image, deep learning methods have shown to work quite well when it is compared to machine learning methods. Therefore, applications that can give meaningful solutions to communication problems are discussed using deep convolutional architectures. These applications were mostly used on the estimation of path loss in the ATG communication systems. In [5], satellite images were used to predict the path loss of a specific Rx in the region. Also, in [16] that deep learning is used for 2D satellite image-based path loss parameters with deep learning, but compared to the technique in this article, a classification-based technique is used instead of a regression. Recently in [4], deep learning with the usage of 2D satellite images have been applied for path loss histogram prediction.

In this thesis, we've worked on two problems using deep learning techniques for communication channel modelling and estimation. We propose a deep learning-based approach on the idea presented [16] for channel parameters estimation as first problem. We aim to analyze the path loss exponent, shadowing factor and their relationships with the 3D model or the height map of a certain region. The contribution that has been done for the first problem can be briefly explained:

- We design a regression network which produces more accurate parameters than using a classification network for channel parameter estimation in [16].
- We prove the ability of deep architectures to learn the complex relationship between wireless network parameters and the height map of a region by using regression method. Therefore, this paper extends the work in [16] by testing height maps as input to deep regression networks for channel parameter estimation.
- We show that better regression performance is achieved when height maps are used compared to use of satellite images.
- We compare two network architectures: a well-known deep network, VGG-16 [8], and a simpler network. Transfer learning [17, 18] is used successfully to fine-tune VGG-16 using a relatively small dataset. The simple network

is trained from scratch. We discuss the performance of these two networks in estimating the channel parameters.

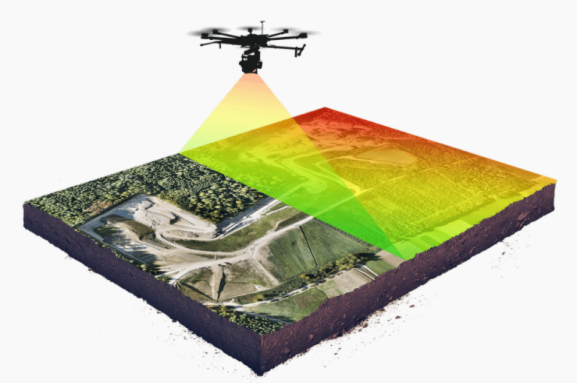
The second problem that we've worked on in this thesis is estimating the excessive path loss point-wise by using deep learning techniques which we think can help more for better network planning. We use a Conditional Generative Adversarial Network (cGAN) to estimate the excessive path loss as point-wise. In [4], fitting a model for path loss histogram prediction was more of a model-based method so our method which is the cGAN usage is more data-based. The novelty of the idea and effective usages are explained as well:

- We used a modified cGAN architecture that has not been used for estimating path loss.
- We show that even if it is not a perfect point-wise path loss prediction but it gives helpful information of the region as a whole.
- We demonstrate point-wise prediction and probability distribution prediction with a high-resolution.
- cGANs can be used to benefit from height maps better than convolutional networks.

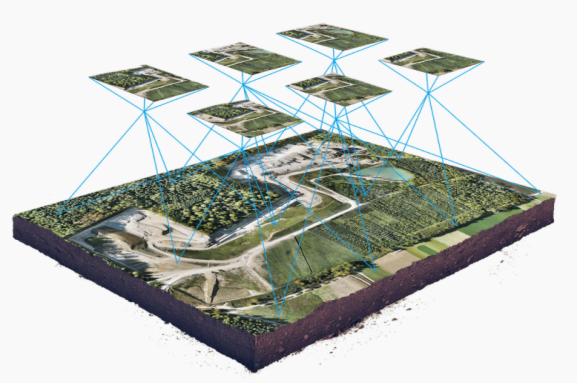
A dataset has been created for both of this problems to train the deep network by using 3D models of targeted areas and running ray-tracing simulations to obtain path loss values and statistical channel parameters.

In this study, we've mentioned that our main goal is to get the wireless channel parameters in the most optimized way for the region and instantaneous needs when it comes to it. In order to do network planning, it is normally necessary to either extract the 3D model of that region and obtain the network parameters through a ray-tracing process, or with actual field measurements. However, with our proposed methods, we aim to make a model that can give the most optimized wireless communication parameters, and this model can be used on a drone or

similar vehicle in the region that has no measurement or simulation knowledge for the targeted area and it can carry out a dynamic network planning. This kind of a scenario is demonstrated in Figure 1.0.1.



(a) Demonstration of 3D mapping for a region.



(b) Extracted 3D map from UAV.

Figure 1.0.1: A drone captures a large number of high-resolution photos over an area that contains elevation/height information [1].

If we go through the scenarios in which this work can be used, it is possible to distribute the density and ensure uninterrupted communication in emergency disaster areas, which can be topographic map of this region by obtaining the heights of the buildings in the settlements by having photogrammetry camera on the drone or similar cameras such as Lidar. With this 3D modeling information obtained from this cameras, height map can be generated and the most optimized channel parameter can be calculated by giving height map information for the region to our model as input. So, with these proposed methods even in the kind

of a scenario that we did not have any knowledge before for the region, we can provide the more optimized communication to users and by this way network optimization can be achieved in a way that can be brought closer to real time processing.

With these approaches of estimating the path loss exponent and shadowing factor, and point-wise excessive path loss with contional GAN using deep learning in our work is relevant to several important issues in communications such as localization, energy-efficient routing, and channel access [19]. By this solutions, we give accurate estimations for the study and design of wireless system.

We present the outdoor propagation models to give a general knowledge about the work that has been done for wireless communications parameter estimation so far in Chapter 2. The first problem, our regression method addressed for estimating path loss component and shadowing factor of the targeted region, dataset generation, and network architecture are explained in Chapter 3. Our second problem that we worked on which is the point-wise excessive path loss prediction with cGAN is also explained in detail in Chapter 4. The simulations and discussions of both of this problems are given in Chapter 5. In Chapter 6, we conclude the thesis.

Chapter 2

Literature Review for Path Loss Prediction Methods

2.1 Empirical methods

Important information about path loss estimation methods for outdoor propagation is presented in this chapter starting with empirical methods. As mentioned before those outdoor propagation models are separated into two which are empirical and deterministic models at first. Okumura model is one of the empirical models that can work between 150-1920 MHz frequencies in urban areas. Since our measurements are done in 900 MHz and 300 meters which is capable of estimating the outdoor path loss with the Okumura model. As well as the frequency, the height of the transmitter needs to be taking into account for the Okumura model and transmitter altitudes should be between 30-1000 meters. To use the Okumura model [20] for path loss calculation, the median path loss is calculated:

$$PL_{dB} = L_F + A_{mu}(f, d) - G(h_{te}) - G(h_{re}) - G_{AREA} \tag{2.1}$$

In this equation where L_F stands for free-space path loss, transmitter altitude gain as $G(h_{te})$, the Rx altitude gain as $G(h_{re})$ and environmental gain as G_{AREA}

such as urban, rural, etc. Furthermore, the equations below are important to calculate the Okumura model's path loss for that environment.

$$G(h_{te}) = 20\log(\frac{h_{te}}{200}) for 1000m > h_{te} > 30m \tag{2.2}$$

Equation (2.2) is used to get the transmitter altitude gain where the transmitter height is between 30 and 1000 meters. If the receiver is placed on less than 3 meters, the equation below has to be used.

$$G(h_{re}) = 10\log(\frac{h_{re}}{3}) for h_{re} \leq 3m \tag{2.3}$$

Also, there is an specific equation where the receiver height is between 3 and 10 meters to get the receiver altitude gain.

$$G(h_{re}) = 20\log(\frac{h_{re}}{3}) for 10m > h_{re} > 3m \tag{2.4}$$

The Hata model is also an empirical model that depends on the localized features in the area under study. Hata model is similar to the Okumura model infrequency wise because it works on 150-1500 MHz and transmitter altitude range at 30-300 meters. Since the Hata model is based on the same formulation of Okumura model, we can compare it with our proposed method and other empirical models. The Hata model equation to estimate path loss for urban areas is provided [12].

$$PL_{dB} = 69.55 + 26.16\log f_c - 13.82h_{te} - a(h_{re}) + (44.9 - 6.55\log h_{te})\log d \tag{2.5}$$

where f_c indicates the frequency (in MHz), h_{te} and h_{re} stands for transmitter altitude and receiver altitude in meters respectively, the distance between transmitter and receiver as d, and $a(h_{re})$ depends on the size of the coverage area for the effective height of the receiver so it changes the environmental basis. Since the receiver is set to 1.5m height the correction factor can be taken as 0 (zero) if the calculations want to be done. For a small and medium cities, $a(h_{re})$ is as follow:

$$a(h_{re}) = (1.1(\log f_c) - 0.7)h_{re} - (1.56(\log f_c) - 0.8) \tag{2.6}$$

while for the large cities:

$$a(h_{re}) = 3.2(\log 11.75h_{re})^2 - 4.97 for f_c \geq 300MHz \tag{2.7}$$

if the carrier frequency is less than 300 MHz, in order to get proper values of coverage area for effective height of the receiver for large cities:

$$a(h_{re}) = 8.29(\log 1.54h_{re})^2 - 1.1 for f_c \leq 300MHz \tag{2.8}$$

As explained in detail in the paper [10] that the Okumura-Hata model can also be modified for suburban and rural areas, and the equations of them will look like below:

$$PL_{suburban} = PL_{urban} - 2[\log(f_c/28)]^2 - 5.4 \tag{2.9}$$

$$PL_{rural} = PL_{urban} - 4.78(\log(f_c)^2) - 18.33(\log f_c) - 40.98 \tag{2.10}$$

Cost-231 Hata [21] model is an extended version of the Okumura model to get more effective results which means that the cover frequency band range extended from Okumura-Hata 150-1500 MHz to 1500-2000 MHz and the formulation of COST-231 Hata is given below:

$$PL_{dB} = 46.3 + 33.9 \log f_c - 13.82h_{te} - a(h_{re}) + (44.9 - 6.55 \log h_{te}) \log d + C \tag{2.11}$$

where C is equals to 0 (zero) for suburban areas and 3 for metropolitan areas.

Another widely used model which is free-space path loss model is also given below:

$$FSPL_{dB} = 20 \log(f_c) + 20 \log(d) + 20 \log(4\pi/_c) \tag{2.12}$$

where the parameter c stands for speed of light in vacuum (m/s). If the values of c and 4π is used in the equation, it will give us a constant value which will be -147.55.

$$FSPL_{dB} = 20 \log(f_c) + 20 \log(d) - 147.55 \tag{2.13}$$

Also, in order to convert d in km and f_c in GHz to meters and MHz, the constant becomes -27.55.

$$FSPL_{dB} = 20 \log(f_c) + 20 \log(d) - 27.55 \tag{2.14}$$

This types of empirical models can give different results depending on the type of area, such as urban, suburban, rural areas when it is correctly configured with their priority parameters.

2.2 Deterministic methods

The way of predicting the outdoor propagation modeling with the deterministic models are more accurate when it is compared with empirical models. Also, empirical models are simpler and getting low prediction accuracy so that empirical models have low performance in the different propagation conditions and frequencies. Ray-tracing modeling is one of the deterministic models that is used for outdoor propagation modeling. Ray-tracing simulation takes a 3D model of the targeted area as input and calculates the receiver signals' propagation characteristics such as delay spread, power, distance to the transmitter, etc. Since the simulations are based on 3D models, if the desired targeted region has no 3D models, we will not be able to use this deterministic model. Unless generating 3D models is an option by using 2D images with photogrammetric measuring method. Wireless InSite (WI) is one of the ray-tracing simulator programs that can calculate the wireless channel parameters from the 3D models. Even if the simulation of wireless channel parameters is very accurate, the usage of this simulation is a time-consuming process. The high performance of predicting the wireless channel parameters from ray-tracing simulations are accurate enough when it is compared to measured ones and can be seen in [2, 22, 23].

The research shows [2] a comparison of practical measuring of wireless channel parameters with relevant simulations of the same area. With the usage of Rohde&Schwarz FSH spectrum analyzer, TS-EMF antenna system, and RFEX software package, with the measurements that are taken in the city of Bosnia and Herzegovina, Banja Luka town. Also, the 3D model of Banja Luka is given into WI ray-tracing software to simulate the same area. The measurements and simulations are made for 900 MHz and 1800 MHz. While the Rx points have been created in a grid of (5m x 5m) at 1.5m height that have 11,500 in the ray-tracing

simulation, in the measurements 80 receivers taken into account that matches with the simulation. Electromagnetic Field Strength (E-FS) is calculated from the measurements for each receiver and results are compared with the simulation from WI software. 50 receivers comparison with measured and simulated results are given in Figure 2.2.1. It can be seen that ray-tracing simulations of a targeted area are able to give satisfactory results when compared with the practical measurements.

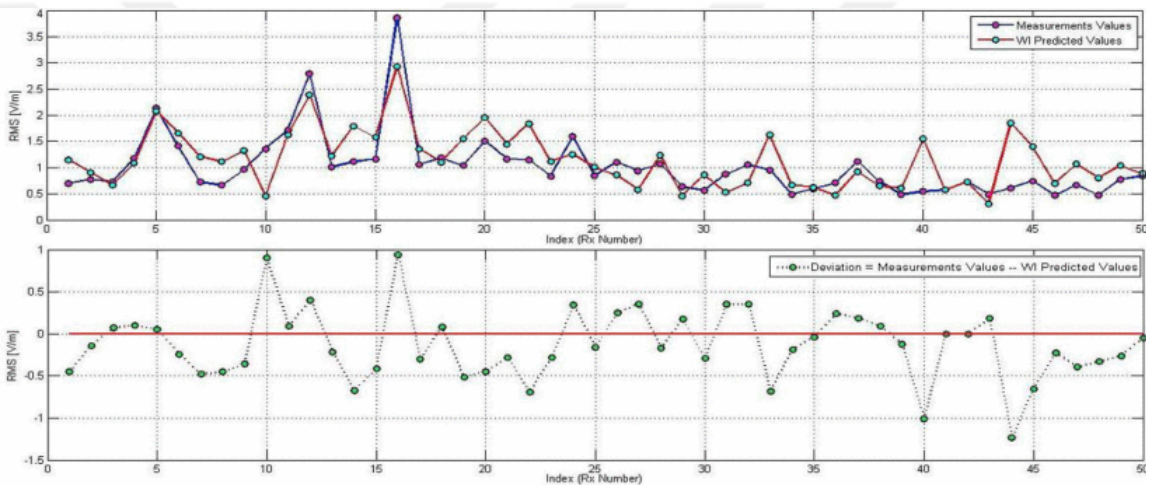


Figure 2.2.1. Comparison of measurement results and software predicted values [2].

2.3 Machine learning methods

Machine learning is also a widely used method in path loss predictions. One of these methods is discussed in paper [3], where the authors estimate path loss and delay spread for ATG communication through machine learning methods such as random forest and K-Nearest-Neighbor (KNN). As is known, machine-learning methods are divided into two categories classification and regression technique. In this machine learning approach, a supervised regression technique was used. A main part of the random forest is a decision tree. Each leaf node in the decision tree represents a category and each inner node represents a test on the feature. Since the learning process of the single decision tree is not effective than

the random forest method because the random forest uses multiple decision trees. Thus, random forests work in two processes; the training examples of each decision tree are selected by bootstrap sampling, meaning that one instance consists of output and corresponding inputs, while the other process is taken from the first set of the feature before the node division of each decision tree, and then selected from the node division property subset. In KNN, predictive samples are compared with ray-tracing samples, and the most similar parent K instances are selected. It is then calculated by the average of observations of upper K samples of path loss and delay spread propagation pre-revaluation values. Since KNN parameters are few, it is a low-cost learning method, but the disadvantage of having a few parameters is that small changes in the parameter lead to large errors. The propagation environment of this ATG communication urban area of the city of Canada, in Ottowa, was taken into consideration to generate the dataset. This dataset generation is made with some little changes compared to other datasets in path loss prediction. The transmitter is placed on a fixed position of 2m above the ground and the receiver is chosen as a UAV device such as a drone to move along the main roads at different heights from 10 to 150m and the position of the UAV recorded every 5m at each. The generated data made for frequencies at 2.4, 5.8, 28 and 37 GHz, and the performances are evaluated for the cases. One of the evaluations of this paper is demonstrated at 2.4 GHz Tx and Rx altitude at 50m in Figure 2.3.1. As it can be seen that empirical models are also compared with the machine learning methods in this case.

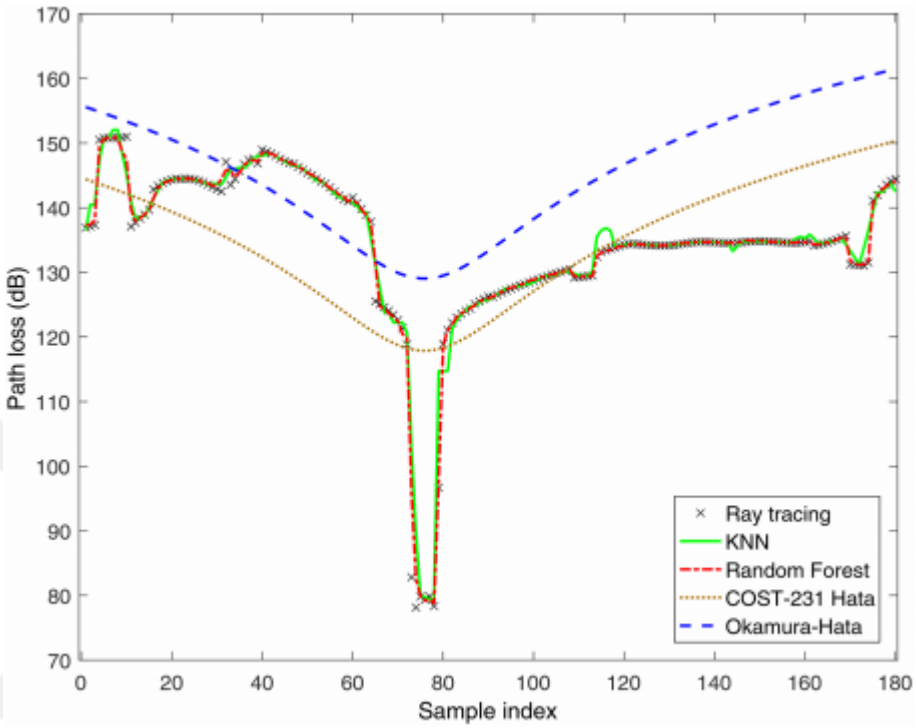


Figure 2.3.1. Machine learning method result for path loss prediction [3].

2.4 Deep learning methods

By the increase of the performance in the Graphical Processing Units (GPU) lead to the use of more artificial intelligent systems. The systems made large usage of deep learning models. Since deep learning models are based on artificial neural networks, we can say that it works like a biological brain, and every information that needs to be processed gets distributed to nodes and it can produce results that can surpass human expert performance. So, the given task to the deep learning model can be done at a sufficient time. In [12], the authors introduced a deep learning model to classify the wireless channel parameters from ATG system using images and implementation steps are similar to our first problem's proposed deep learning model which will be explained in detail. With the usage of ray-tracing simulation from WI at 900 MHz frequency and transmitter height at 300

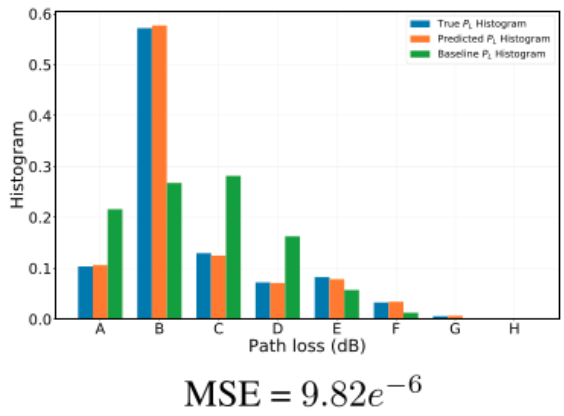
meters, a dataset is generated. This dataset contains satellite images, corresponding 3D models, and wireless channel parameter values that are computed from the widely used large-scale path loss model formula for each satellite image by using the simulation values. Then, satellite images are given to the deep learning model that is established to classify the wireless channel parameters (n,σ). By this method, the classification performance stated as %86 for n and %76 for σ which is accurate enough for the targeted regions.

In one of the recent works in [4], the authors provided a reliable approach of deep learning method to predict the histogram of path loss distribution for the targeted area at different frequencies such as 900 MHz, 3.5 GHz, and transmitter heights at 40, 80, 300 meters. This approach is used with bin-wise histogram path loss prediction, meaning that using 8-bin clustered for path loss in dB to predict it using satellite images with implemented deep learning model. Therefore, 8-bin representation and one of the sample results with the targeted area is given in Figure 2.4.1. The performance of this method is evaluated by using a Mean Squared Error (MSE) histogram-wise. The targeted area's MSE between predicted path loss histogram and true path loss histogram is shown as well in the Figure below (c).

$PL(d_0) = 63.44\ dB$	
A	$PL - PL(d_0) \leq 20$
B	$20 < PL - PL(d_0) \leq 30$
C	$30 < PL - PL(d_0) \leq 40$
D	$40 < PL - PL(d_0) \leq 50$
E	$50 < PL - PL(d_0) \leq 60$
F	$60 < PL - PL(d_0) \leq 70$
G	$70 < PL - PL(d_0) \leq 80$
H	$80 < PL - PL(d_0)$

(a) 8-bin representation

(b) Targeted area



$\text{MSE} = 9.82e^{-6}$

(c) Histogram path loss prediction of targeted area

Figure 2.4.1: 8-bin, satellite image, and result of histogram path loss prediction [4].

Another method used after recently the usage of satellite images to predict path loss or parameters using deep learning models is given in the [5]. The proposed deep learning model has a simple Deep Neural Network (DNN) architecture that is combined with a Convolutional Neural Network (CNN) and Neural Network (NN) layers. The general demonstration of the proposed architecture is shown in Figure 2.4.2. It can be seen that not only the deep learning model is getting trained, also the path loss model that is defined as $L(d)$ is used for assisting the learning process. $L(d) = PL(d) + G_{t_x}$, where G_{t_x} is the estimated transmission power and related gain. Thus, the authors claim that the proposed simple DNN model is capable of improving path loss prediction at unseen locations for 811

MHz with 1 dB and 4.7 dB for 2630 MHz. The effectiveness of this work is shown with the targeted and predicted results for 811 MHz and 2630 MHz is given in Figure 2.4.3.

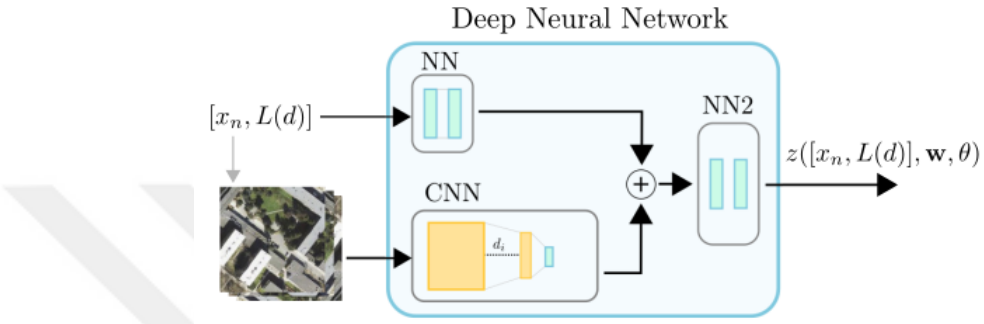
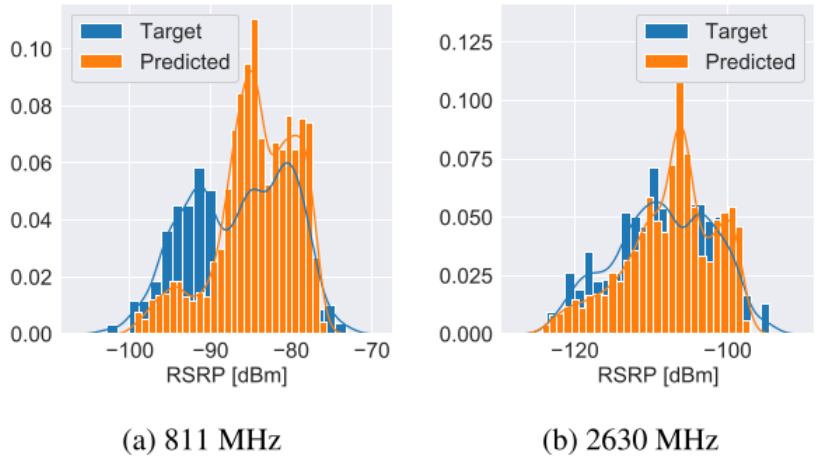


Figure 2.4.2. DNN model architecture from [5].



(a) 811 MHz (b) 2630 MHz

Figure 2.4.3. Results from the paper of [5] in RSRP.

2.5 Generative adversarial network methods

With the increase of deep learning methods rapidly, new applications occurred such as GAN methods. The idea of GANs was first introduced in [24]. The basic principle of GAN is to approximate the unknown distribution of a particular dataset by optimizing an objective function through a game between the generator

and the discriminator [25]. It is stated that GAN can update dynamically through training of the discriminator which means that the generator's aim to fool the discriminator to get targeted results. The total loss consists of generator loss and also from the discriminator loss, whose function is to determine how far the output is from the real input. The GAN applications are mostly known as hard tasks which means that it is difficult to train a convenient model. There is a different kind of GAN variations which are used for style transferring [7], semantic imaging [26], producing a super-resolution images [27], image to image transferring [6], etc. Sample results for these types of GANs shown in the Figure 2.5.1. Most of the GANs produce images as an output but some works have been done on producing synthesized speech with GAN like in the paper [28]. Besides the GANs, a cGAN is a type of GAN that simply can give the targeted output (Y) to the generator and discriminator [29]. Thus, image generation can be conditional on a class label that it is possible to use on different approaches such as path loss predictions. Since this idea of predicting point-wise path loss with cGAN is novel, there are not many applications in this area so our GAN based proposed method is explained in detail in Chapter 4.

(a) Image to image transferring

(b) Style transferring

Figure 2.5.1: Sample results of GANs [6, 7].

Chapter 3

Regression of Path Loss Parameter Prediction Model

Our interest in this thesis is the outdoor wireless channel parameter estimations, and the first problem that we worked on is explained with our proposed method in detail. New approaches have emerged in wireless communication applications with the rise of machine learning and deep learning techniques in various fields [30]. Predicting the path loss by using machine learning techniques are declared in [31] and [32] for areal path loss prediction. Also, the study in [33] and [34] show that by using the height of Unmanned-Aerial-Vehicle (UAV) to get the characteristics of the wireless communication parameters vary with different heights when it is predicted with machine learning methods at the low-frequency band. Aside from the usage of machine learning methods to estimate path loss, the computational and visualization capabilities of computers have accelerated recently so the deep learning method usage increased rapidly. The wireless technology identification using CNN is given in [35] and by using 2D satellite images to predict path loss components in [16], and predicting the histogram of path loss distribution in [4] can be given as examples of deep learning usages for wireless channel parameter estimation methods that has been mentioned before in the literature review.

Propagation models are mostly used to estimate the mean signal power from the transmitter to the receiver. Also, any distortion that happens to the signal between the transmitter and receiver is known as path loss. In the proposed regression model, we try to predict the path loss component and shadowing factor so with the usage of the large-scale free space path loss model [16]:

$$PL(d) = PL(d_0) + 10n \log_{10} \left(\frac{d}{d_0} \right) + X_\sigma, \tag{3.1}$$

where the d at $PL(d)$ is a distance and $PL(d)$ is the path loss at an explicit point from the transmitter. Also, d_0 of $PL(d_0)$ is the reference distance, and $PL(d_0)$ is reference path loss. X_σ is a random variable whose mean is zero and the standard deviation of X_σ is σ. In [36] points out that σ is a variable that indicates the extent of shadowing. Model (3.1) is commonly used for the UAV communications estimation [37]. In this work, we mainly focus on estimating the path loss exponent (n) and shadowing factor (σ) from height map of a given region as well as satellite image. As an input to the deep neural networks height map and satellite images are given and wireless channel parameters (n, σ) are predicted.

3.1 Dataset generation

Dataset generation is the key component to the training for deep learning purposes. In this study, satellite and 3D building models and various urban areas were selected. SketchUP[1] and 3D models were used in the auxiliary PlaceMaker[2] program to obtain this data. This data covers (1.8 x 1.8) km of New York City in each image. Ray-tracing simulations were made with Wireless Insite[3] program, which works using 3D building models. In this simulation, the receivers are simulated to be 1.5 meters above the ground with 12,100 (110 x 110) points in the

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

form of a particular grid and the transmitter is placed 300 meters high in the middle of the specified grid. Only outdoor receivers are used. Transmitter power and antenna have been set to 60 dBm and omni-directional respectively. Table I demonstrates the simulation parameters in detail. As a result of the simulation, power levels were taken for each receiving point, and path loss was calculated. For channel parameter calculation (n, σ), the least squared method is used and described by the equations below. We can see equation (3.1) translated into vector form in equation (3.2). When we accept X_σ as 0 (zero) and solve the equation, we can get the value using equation (3.3).

$$Y = nA + X_\sigma, \tag{3.2}$$

If we define $10\log(\frac{d^{(i)}}{d_0})$ as A and $(PL - PL_0)$ as Y vectors, we can calculate the n parameter with the following equations (3.3), (3.4), (3.5).

$$n = (A^T A)^{-1} A^T Y, \tag{3.3}$$

$$n = \frac{\sum_{i=1}^{N} Y_i A_i}{\sum_{i=1}^{N} (A_i)^2} \tag{3.4}$$

$$n = \frac{\sum_{i=1}^{N} (PL(d_i) - PL(d_0)) \left(10 \log \left(\frac{d_i}{d_0}\right)\right)}{\sum_{i=1}^{N} \left(10 \log \left(\frac{d_i}{d_0}\right)\right)^2} \tag{3.5}$$

As shown in equation (3.6) that we can find this shadowing factor variance (σ) by using the computed path loss exponent (n). Then, we can get the (σ) which is the standard deviation of X_σ.

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} \left(PL(d_i) - PL(d_0) - 10n \log \left(\frac{d_i}{d_0}\right)\right)^2 \tag{3.6}$$

Table 3.1: Simulation Parameters

Parameters	Values
Transmission Frequency	900 MHz
Transmit Power	+60 dBm
Transmitted Signal	Sinusoid
Transmitter Antenna Height	300 m
Receiver Antenna Height	1.5 m
Antenna Polarization	Vertical
Antenna Radiation Pattern	Omni-directional
Bandwidth	8 MHz
d_0	57.28 m
$PL(d_0)$	63.44 dB

Channel parameters (n and σ) have been added to our dataset with each satellite image that is intended to be used.

To make wireless channel parameter estimation, good results have been achieved with satellite images using deep learning [16], and it is targeted to use height maps to further improve this method. We reveal that the detailed altitude data of the height maps provide more detailed information than the information obtained from the satellite images in deep learning networks. In Chapter 5, we discuss the performance of using height maps and satellite images. It can be seen in the block diagram in Fig. 3.1.1 that the detailed production process of the dataset is described.

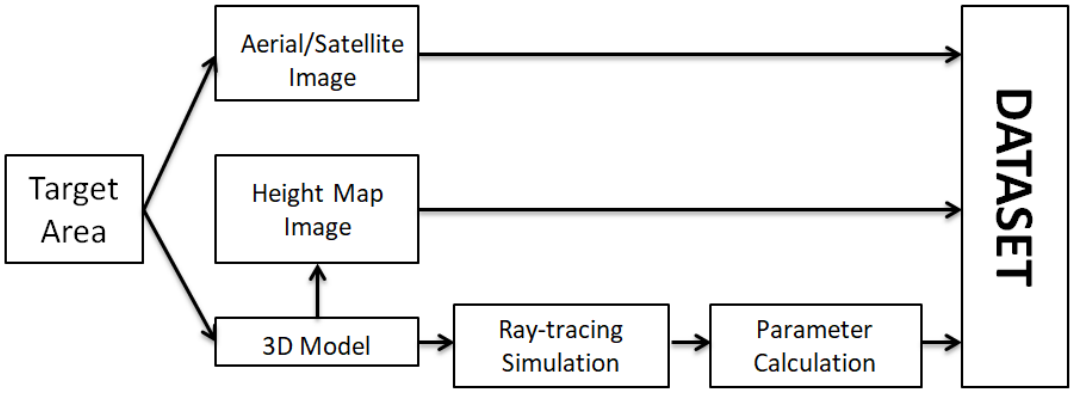


Figure 3.1.1. Block diagram of the dataset generation.

3.2 Height map generation

Making an useful and proper dataset is a hard task so in order to create such dataset, height map generation can be explained by the following main steps,

- Converting 3D files to open by MeshLab[4].
- Using Stanford Triangle Format in CloudCompare[5] to generate height maps.
- Using Actiona[6] to automate height map production.
- Coordinate system transformation for receiver locations to height maps coordinate system
- Making Wavefront 3D Object File (.OBJ) files to visualize powers in 3D with CloudCompare
- Generating receiver locations images
- Cropping from boundaries of receiver locations for height map and resizing
- Storing power information file in pixel wise
- Intensity mapping
- Wrapping operation to get identical with satellite imagery

As mentioned in the dataset generation part that different regions are selected for satellite images and their 3D building models. Those selected area's 3D models are in Collada (.DAE) format to open and obtain the information inside a 3D files, the format of the 3D models needed to be changed to Stanford Triangle Format (.PLY) format by using the program called "MeshLab". This operation has to be done because the information is needed for every 3D building where X

[4]http://www.meshlab.net/
[5]https://www.danielgm.net/cc/
[6]https://www.howtoinstall.co/en/ubuntu/xenial/actiona

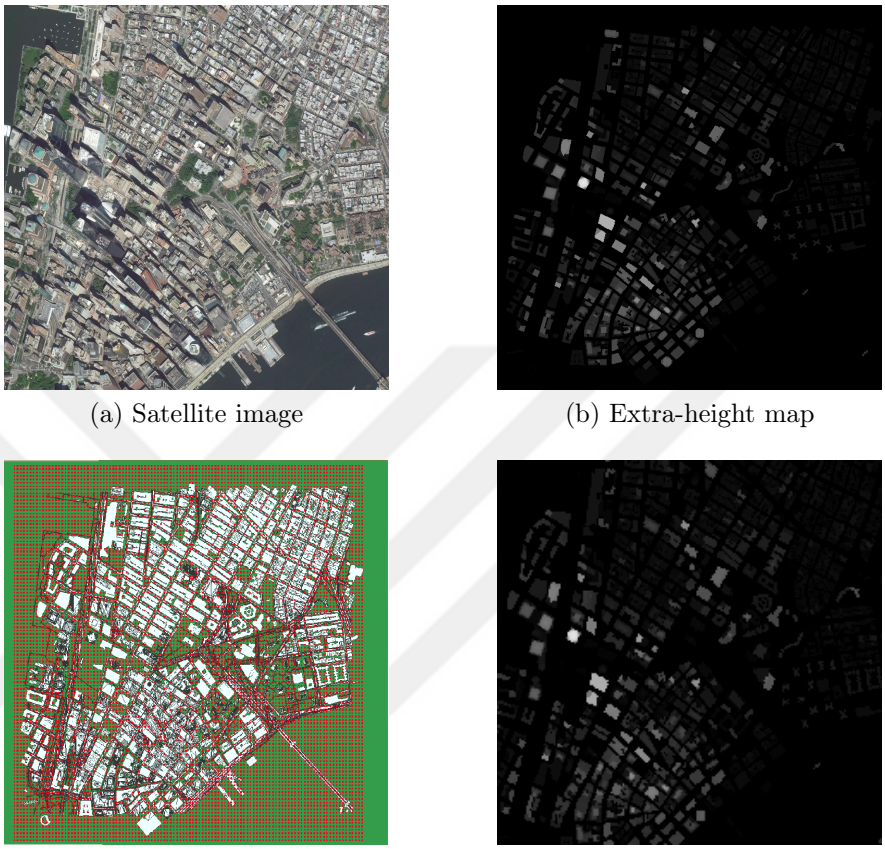
stands for horizontal coordinate, Y for vertical coordinate, and Z for the height values.

After having the necessary information from the 3D format, the Stanford Triangle Format is opened by CloudCompare, and the information of Z(height) gray-colored where the surface is black and the highest building is white. The gray-colored images named as "Height map" are generated and saved as a bitmap. The bitmap can store the color data of each pixel in the image without any comparison applied and in this way, the exact level of height information can not be lost.

Since generating a height map is a single-time process if we want to make it as a large dataset the process needs to be repeated for the rest of the 3D files. To make the process quick and productive, the "Action" automation tool is used. It is a simple tool that repeats the same steps to generate height maps.

The generated height maps from 3D building model files have more information that is compared with the satellite images, it can be seen in Figure 3.2.1 (b). Thus, extras of the image have to be removed to be the same as the satellite image in Figure 3.2.1 (a).

We have used it to simulated our 3D models and to get power distribution for each receiver location on the 3D model where the transmitter is placed in the middle on the WI program. After getting the simulation results from WI, the power files are used. The power file includes X-Y coordinates and power values for each receiver point but the coordinates of the powers do not match with the coordinates of the 3D model when we open the 3D visualization of the power as .OBJ file in CloudCompare the reason behind is that WI and CloudCompare programs are not using the same global coordinates so, to match the powers from WI with the height maps, we use the receiver location images from WI and get the ratio between the buildings and the receiver location(red points). A receiver location image from WI is shown in Figure 3.2.1 (c), the red points indicate the receiver locations. Moreover, we needed to shift the X-Y-coordinates of power so that the geometric center coincides with the cartesian origin(-1090 from X and



(a) Satellite image

(b) Extra-height map

(c) Receiver location image

(d) Height map

Figure 3.2.1: A sample satellite, extra-height map, receiver location image and height map.

Y). We use the ratio to scale the X-Y coordinates of power. After scaling is done, it is needed to fix the shifting of the receiver locations to centralize all of the images like WI did. So, we found out that if we added 4.84 to X and 0.5 to Y coordinate, it exactly matches the WI configuration. Consequently, the simulated powers and the height maps are matched.

After matching the powers with height maps, we need to find the receiver location images' boundaries and from these boundaries height map images will be cropped to be the same as the satellite image. However, we need matched power and height maps to be resized to (655,655) because we have (110 x 110) receiver locations and we will need to show it pixel-wise on a regular grid. Since we have 12,100 (110 x 110) receiver locations that consist of powers and it has to

be distributed on a regular grid so we thought if we can make 1 power pixel and 5 empty pixels on a grid, it will be fine for the rest of the receiver locations. In this way, we were able to assign the receiver powers for pixels. Height map shows the height of the buildings as intensity for the pixels in the building area, where the values are normalized from 0 to 255 which is named as "Intensity Mapping" process. Each intensity corresponds to a fixed height (with a small margin of error due to quantization) across the whole dataset. The highest building, which is the Empire State building, has a height of 381 meters and its pixel intensity is set to be 255. Every other building height is normalized accordingly.

The satellite images are not ortho-rectified which is not identical for the height maps so that we have used perspective warping operation. In this operation, we used 8 main points that can set-up the homography matrix and applied it to satellite images. The last version of the satellite image and height map can be seen in Figure 3.2.1 (a) and (d). Consequently, we will end up having resized, cropped and intensity mapped height maps that exactly matched with satellite images and their simulation information.

3.3 Network architecture

When we look at deep learning architectures, some architectures achieve sufficient results through deep learning architectures and regression analysis. One of these architectures is VGG-16 [8] given in Fig. 3.3.1. This architecture is used to estimate path loss exponent and shadowing factor parameters. The VGG-16 architecture consists of 13 convolution layers and 3 fully connected layers and has 134 million trainable parameters. The VGG-16 network is trained by using a transfer learning technique that is pre-trained using a 1000-category ImageNet dataset, which was implemented using ImageNet's weights except for the last fully-connected layer. It is proven that the transfer learning technique has a great effect on the results shown in [38] as well because image features benefits from well-defined filters of the early convolutional layers so they get more effective results than training from scratch. As mentioned above that only the last layer

which is the output layer of the VGG-16 is modified to give a single output, and sigmoid is used as an activation function. For the training loss, MSE is preferred. As an optimizer Stochastic Gradient Descent (SGD) is used with a momentum of 0.9 and the learning rate is chosen as 0.0001.

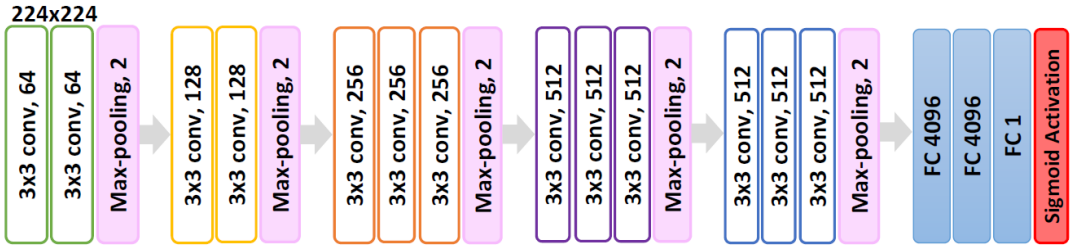


Figure 3.3.1. VGG-16 architecture [8].

A low-complexity network is also used to analyze the effect of shallower architectures on the prediction of the wireless communication parameters. The network is designed upon a sample path loss prediction architecture in paper [5]. This structure contains batch normalization and max-pooling layers for each 6 convolutional layers. The architecture ends up with 3 fully connected layers that have 264,000 trainable parameters. Adam optimizer and 0.001 learning rate are used to perform the training from scratch, i.e. with random initialization.

We analyzed the impact of the less complex shallow architecture for the estimation of wireless communication channel parameters. This network is designed to be a modified version of a network previously used [5] for path loss estimation. This structure is designed with 6 convolutional layers followed by batch normalization and maximum pooling layer, and 3 fully-connected layers at the end. This low-complexity network contains 264,000 trainable parameters, with its optimizer Adam and learning rate at 0.0001, but it is trained from the scratch, unlike the VGG-16 network that we modified.

Our data is divided into %75 (725 images) for training and %25 (241 images) for testing to be normalized and trained in the [0-1] range, including the wireless channel parameters (n and σ). The data augmentation method is just used for the training data. This data augmentation method is used on the original samples (i.e. satellite images and height maps) of training data with flipping vertically /

Figure 3.3.2. Low-complexity architecture adopted from [5].

horizontally, rotating left by 90-180-270 degrees, and also flipping vertically and rotated by 90-270 degrees. This data augmentation technique [39] is used because of reducing the over-fitting as well as to increase the number of training samples to make the network learn better. Also, the *Keras* deep learning framework is preferred in this work.

Chapter 4

Point-wise Excessive Path Loss Prediction with Conditional GAN

Path loss is also important for wireless channel characteristics as much as the path loss exponent and shadowing factor. Path loss can be calculated simply by using the Tx and Rx powers for ATG UAV communications and also can be seen by the formulation below:

$$PL_{dB} = P_{T_x} - P_{R_x} \tag{4.1}$$

By the usage of deep learning with height maps brought us the idea of using cGAN to predict the point-wise path loss as a second problem in this work. The dataset that is generated for this purpose contains mostly urban and suburban areas so predicting the excessive path loss is more logical in this case. The simple demonstration of free-space path loss and excessive path loss is given in Figure 4.0.1 for ATG communication.

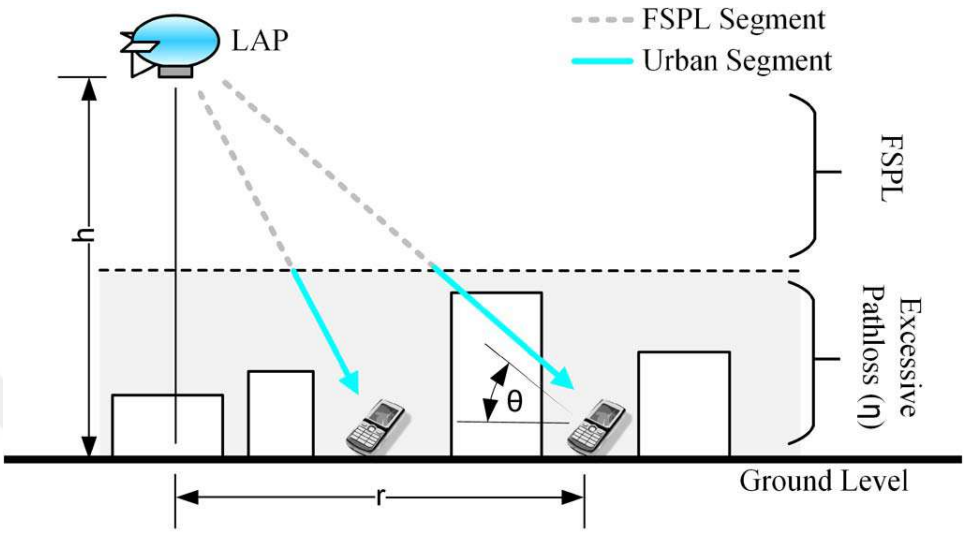


Figure 4.0.1: Air-to-ground excessive path loss example. [9].

It can be understood that signals get affected by the obstacles in front of them such as buildings and this situation is named as Line-Of-Sight (LoS). When there is a LoS, the excessive path loss is higher than in Non-Line-Of-Sight (nLoS) cases but even if there are no obstacles affecting the signal, it can get strong reflection and refraction like mentioned in the paper [40]. Eventually, excessive path loss is calculated for our dataset. The calculation of excessive path loss formula η is given in (4.2):

$$\eta = PL_n - FSPL \tag{4.2}$$

PL_n is the path loss values for the number of receivers and $FSPL$ formula is given in (2.14).

4.1 Conditional GAN dataset generation

Conditional GAN dataset generation has almost the same as the dataset generation in Chapter 3. The dataset which was explained in Chapter 3 is made for frequency as 900 MHz and transmitter altitude as 300 meters but in this problem, it has been decided to use the same simulation parameters except for the transmitter antenna height so we changed from 300 meters to 80 meters when making the dataset. The reason for lowering the transmitter antenna height is to see whether our model can predict true values of excessive path loss for an area. The work is done point-wise which means that for each receiver in the image, the excessive path loss is calculated. After getting the true values for excessive path loss for every receiver for each image in the dataset, it has been decided to make excessive path loss image where the true path loss values are placed on an uniformly distributed rectangular grid. To make path loss image, it needs to be matched with the size of the input image (height map, satellite image) which is (256x 256) so the path loss image is resized from (110 x 110) to (256 x 256) by using nearest interpolation. Since, this work is done only for outdoor propagation, we removed the path loss values that are indoor and sample of excessive path loss image can be seen in Figure 4.1.1.

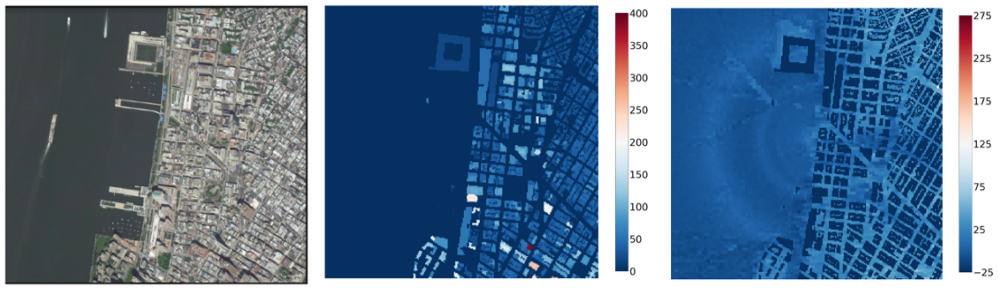


Figure 4.1.1: Satellite images, their corresponding height map, where the colorbar is in meters, and path loss, where the color bar is in dB images are shown for a certain region.

4.2 Conditional GAN network and hyperparameters

Conditional GANs can perform tremendously in a large number of applications, and key application areas are usually style transfer, image super-resolution, etc. One of the largest uses of the conditional GAN network, pixel-to-pixel [6], has been modified and used to predict point-wise excessive path loss and explained in detail in this section. To explain our modified conditional GAN network well, we chose to tell it part by part. Our proposed cGAN network consists of the generator and discriminator part. The generator part is described firstly, the modified U-NET architecture is used as a generator in our cGAN. The U-NET architect consists of encoder and decoder parts so that the same input size image is expected in the output and by this method, the input image is transformed to get the targeted image.

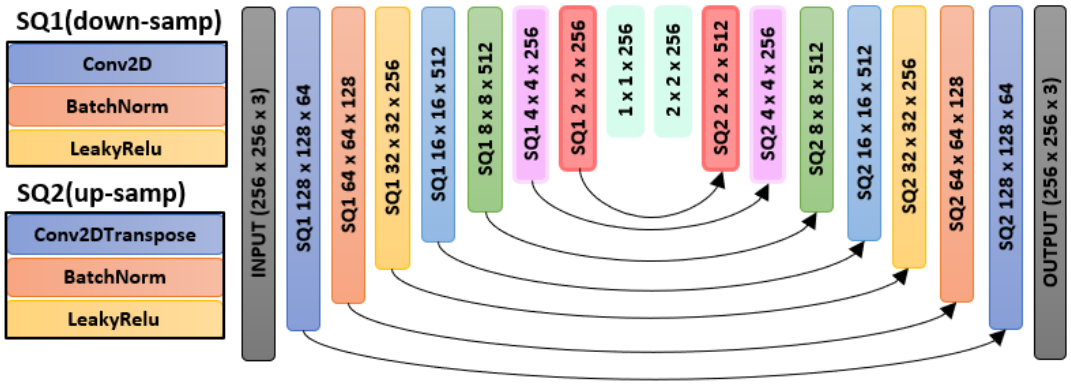


Figure 4.2.1: Generator's architecture. SQ1 has conv2D layer with strides (2) and padding (same). Sequential 2 (SQ2) has conv2DTranspose layer with stride (1) and padding (same).

The generator architecture consists of 18 convolutional layers with skip connections that can get valuable information from encoder to decoder and it is indicated as arrows in Figure 4.2.1. Each block in the encoder contains convolutional, batch normalization, and leaky relu layers. Also, each block in the decoder has transposed convolutional, batch normalization, and relu layers. Since

the generator architecture in detail is shown, the training of the generator part is also given in Figure 4.2.2. As it can be seen from the block diagram that the generator takes input imagery which can be a satellite image or height map. The generated image from the input given to discriminator and as well as the input image, discriminator with the taken input imagery learns to classify fake and real path loss image by the usage of binary cross-entropy so that generator's real purpose is can be seen as fooling the discriminator.

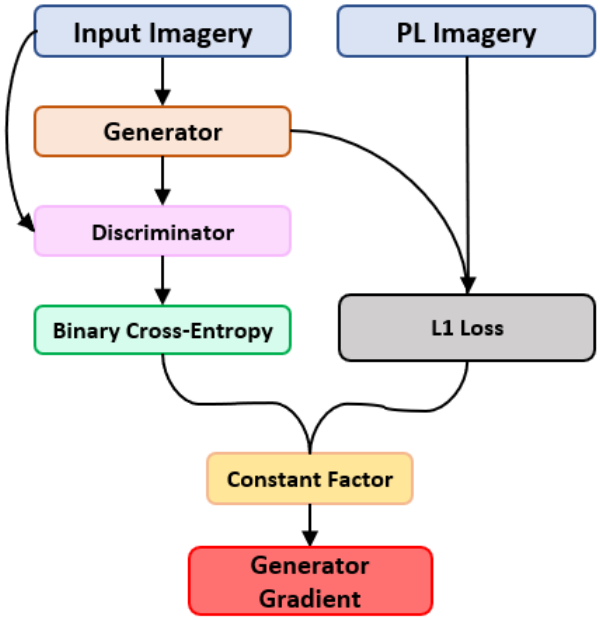


Figure 4.2.2: Training of generator.

Our network needed a loss function that can work on the outdoor propagation so the generator loss function is established on just outdoor loss. Outdoor loss means that the structural areas such as buildings were not considered as values that can effect on the getting generator loss. Then, from the values of outdoors, the loss function is chosen as L1 Loss and the formulation is giving below:

$$L1_{loss} = \sum_{i=1}^{n} |y_{true} - y_{predicted}| \tag{4.3}$$

The benefit of using L1 loss is to minimize the error between true and predicted. Another reason is, the L1 loss is not simply affected by the outliers of the data. Consequently, we get the generator gradient by these steps in order to get better-generated images.

As is known that discriminators try to separate the actual data from the data generated by the generator. Discriminator architecture is inspired by Patch-GAN by the paper in [41]. It is called PatchGAN because patch-by-patch is taken into account to penalize the structure instead of taking the data as a whole. In Figure 4.2.3 the discriminator gets 2 inputs which are input and path loss images that's why the input size is (256 x 256 x 6). After the input layer, the following layers are indicated like generator SQ1 which are the down-sampling layers and continues with the block part that consists of (conv2D-BatchNorm-LeakyRelu) and ends until the discriminator output reaches the shape of conv2D (5 x 5 x 1) which is called discriminator receptive field.

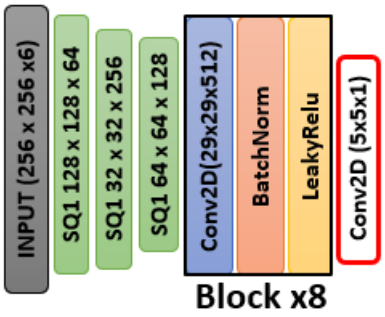


Figure 4.2.3: Discriminator's architecture. There are eight times of the layers in shown in the block and conv2D has stride as (1).

Therefore, the discriminator can look at the part of the image (5 x 5). We have found out that making the discriminator's receptive field smaller gives us better path loss estimations so that the most suitable receptive field for our purpose is chosen as (5 x 5). The training of the discriminator is shown in Figure 4.2.4. The loss of discriminator takes 2 inputs as mentioned before; real images and generated images. The real loss is calculated by using the binary cross-entropy loss of the real images which are an array of ones and generated loss is also calculated by

generated images using binary cross-entropy loss which is an array of zeros. Then, the summation of real loss and generated loss form the total discriminator loss. At last, the discriminator updates its weights by back-propagation from the total discriminator loss.

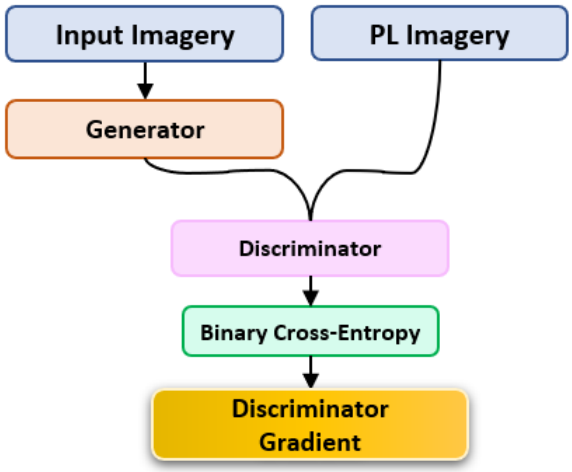


Figure 4.2.4: Training of discriminator.

To get the adversarial loss, the network tries to figure it out how many times the generator failed to deceive the discriminator so that the output of the discriminator for generated images should be ones by the output of binary cross-entropy but it sometimes fails so getting the difference between real and generated image gives our total conditional GAN loss. In this work, the data is divided into %70 (700 images) for training and %30 (300 images) for testing to be normalized and trained in the [0-1] range. Also, $Tensorflow$ deep learning framework is preferred. Adam as an optimizer, 0.0002 as learning rate is chosen for both generator and discriminator network. We trained on Tensorflow 2.0 on Nvidia RTX 2080 Ti GPU. The training takes about 12 hours for 500 epochs. The proposed cGAN network is trained from scratch.

Chapter 5

Simulations and Discussions

5.1 Regression model performance for (n,σ) prediction

Table 5.1: Path Loss Exponent (n) Prediction

	Mean Squared Error (MSE)		
Architecture	Satellite image	Height map	Variance
VGG-16 [8]	0.46×10^{-2}	0.41×10^{-2}	0.65
Low-complexity [5]	0.97×10^{-2}	0.62×10^{-2}	0.65

Table 5.2: Shadowing Factor (σ) Prediction

	Mean Squared Error (MSE)		
Architecture	Satellite image	Height maps	Variance
VGG-16 [8]	**5.28**	5.43	90.63
Low-complexity [5]	4.11	**3.41**	90.63

In this section, we demonstrate different types of deep learning models and their performances on the wireless communication channel parameter dataset that has been made while this work is being done. Classification-based prediction modeling problems differ from regression modeling problems. Since this work is done upon regression modeling problem, MSE is used between predicted and true

values that needed to be analyzed for satellite images and height maps. In Tables 5.1 and 5.2, the results are given for path loss exponent (n) and shadowing factor (σ). The variances are provided with a table for a better comparison of the test dataset. When we look at the ratio of MSEs in (n and σ) to the variation, it proves that the regression network is an accurate predictor in channel parameters. n is better predictable compared to σ by all tested networks and it can be seen from Figures 5.1.1 and 5.1.2 like in Tables 5.1 and 5.2. The σ parameter is more difficult to predict than n because it is associated with uncertainty and noise level in path loss data. Figures 5.1.3 and 5.1.4 are presented as actual values (sorted in ascending order) against estimated values for scenarios tried with different networks. We can observe in both the satellite images and height maps that the σ values have some outliers in other words mispredicted values.

When looking at the tables where satellite images and height maps are compared, the estimation of height maps with the VGG-16 network results better, while only the sigma parameter's estimation with the VGG-16 network results poorly. It was understood that the height information was necessary for the correct modeling of the wireless channel and this information could not be obtained from satellite images.

We get better results for the n values of the VGG-16 network and much less results for σ values when VGG-16 and low-complexity network's performances are compared. Since the size of the training data is limited, it causes over-fitting on high-complexity models, such as on the VGG-16 network. This over-fitting negatively affects the test set because it learns about the noise in the σ values in the training data, which explains why the performance value of the σ value is low in its estimation. In the case of VGG-16, it is observed in Figure 5.1.3 and 5.1.4, where there are many outliers for both satellite images and height maps. We see that a low-complexity network for σ achieves better results. This is because the low-complexity network does not suffer from over-fitting and has produced better results for σ than the VGG-16 network. For both satellite images and height maps that various test sample results of this study were given in Figure 5.1.5.

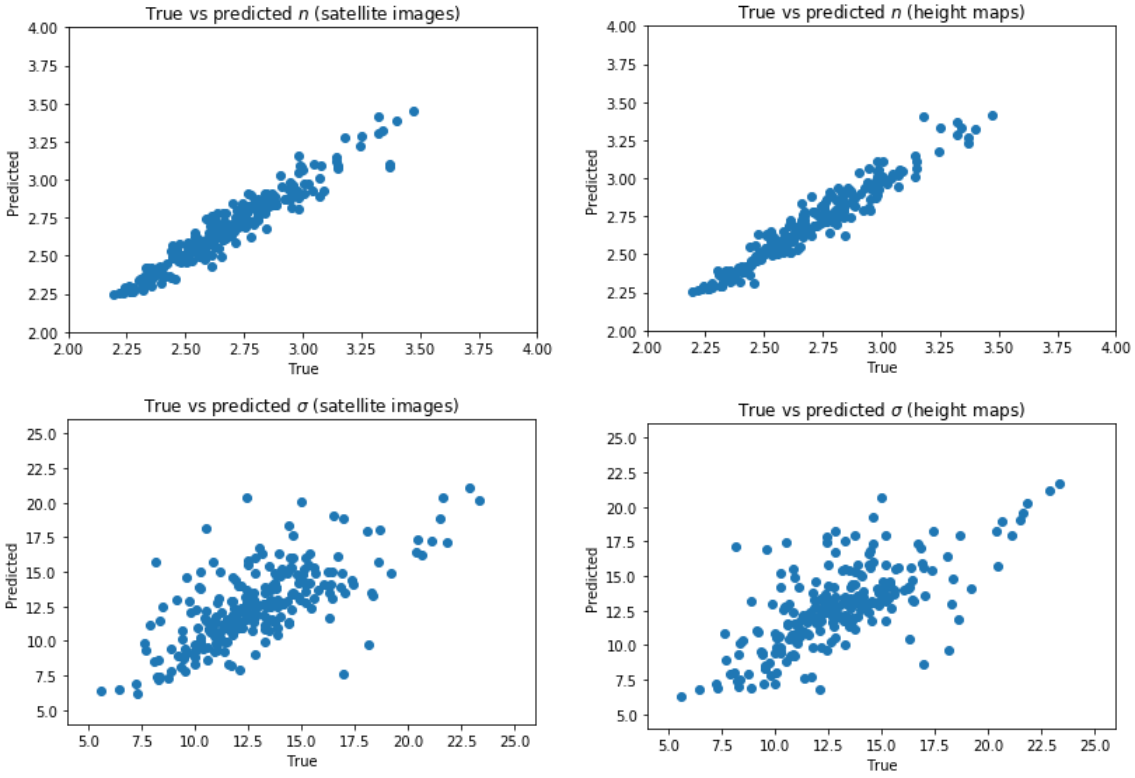


Figure 5.1.1. VGG-16 architecture [8] true vs. predicted scatter plots for test data.

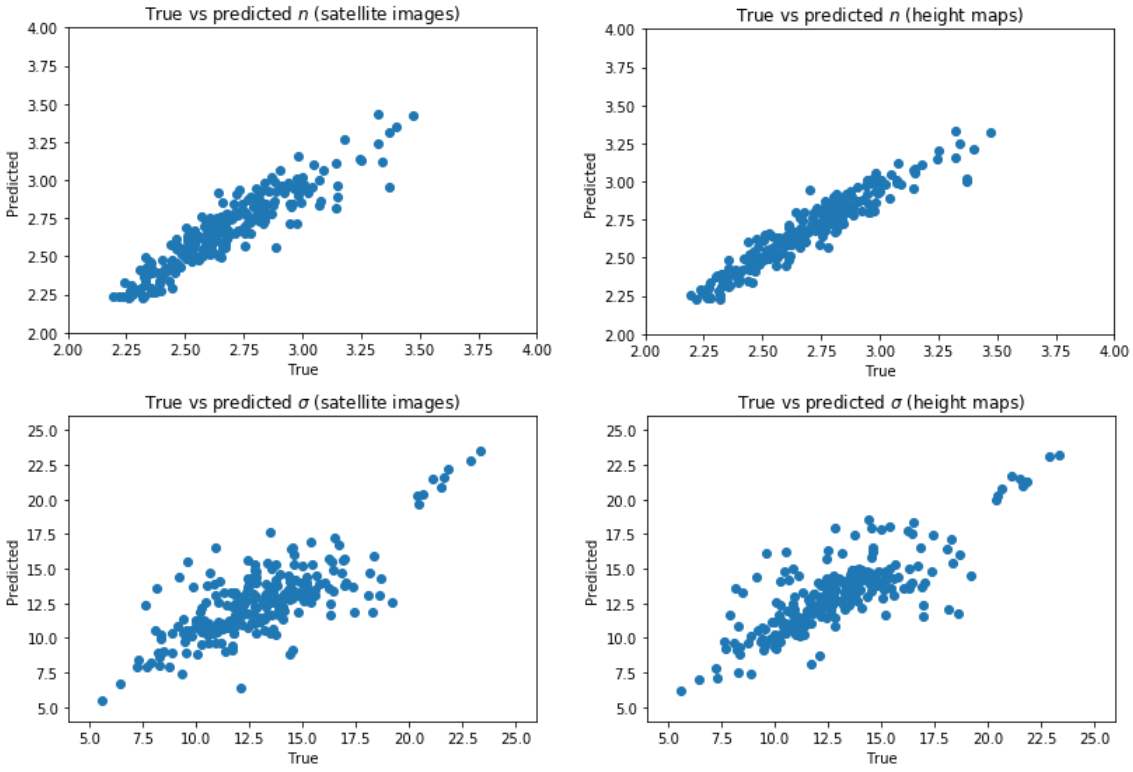


Figure 5.1.2. Low-complexity architecture [5] true vs. predicted scatter plots for test data.

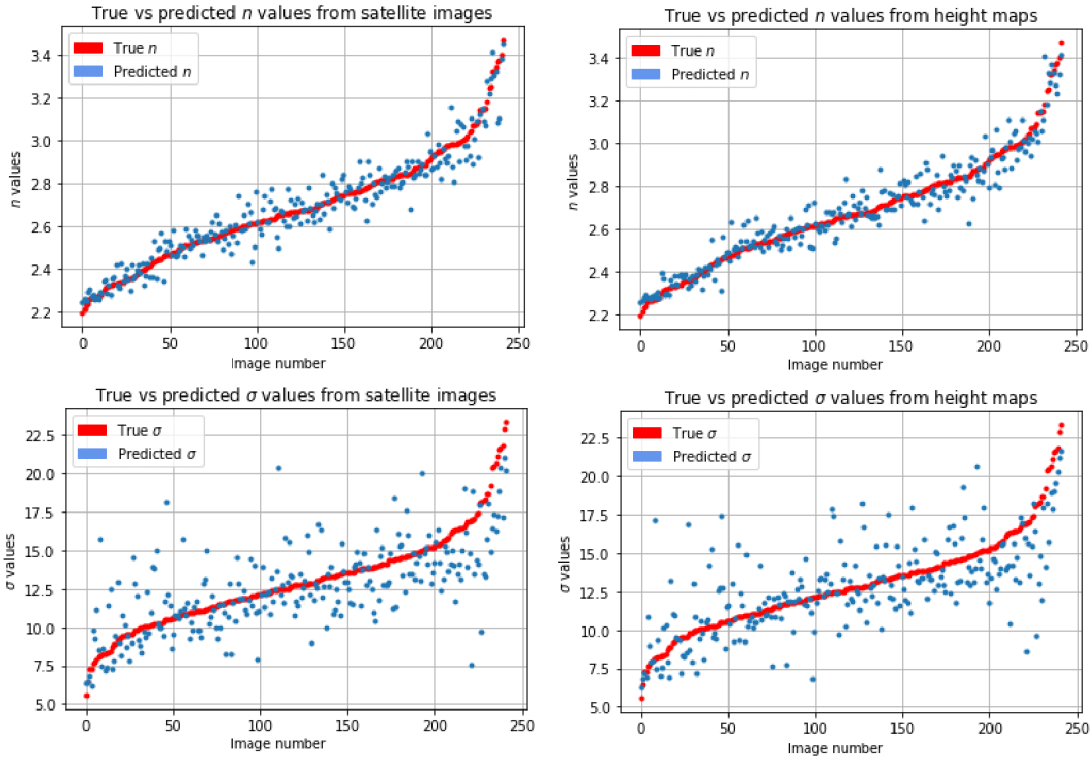


Figure 5.1.3. Prediction of test samples based on the VGG-16 architecture [8].

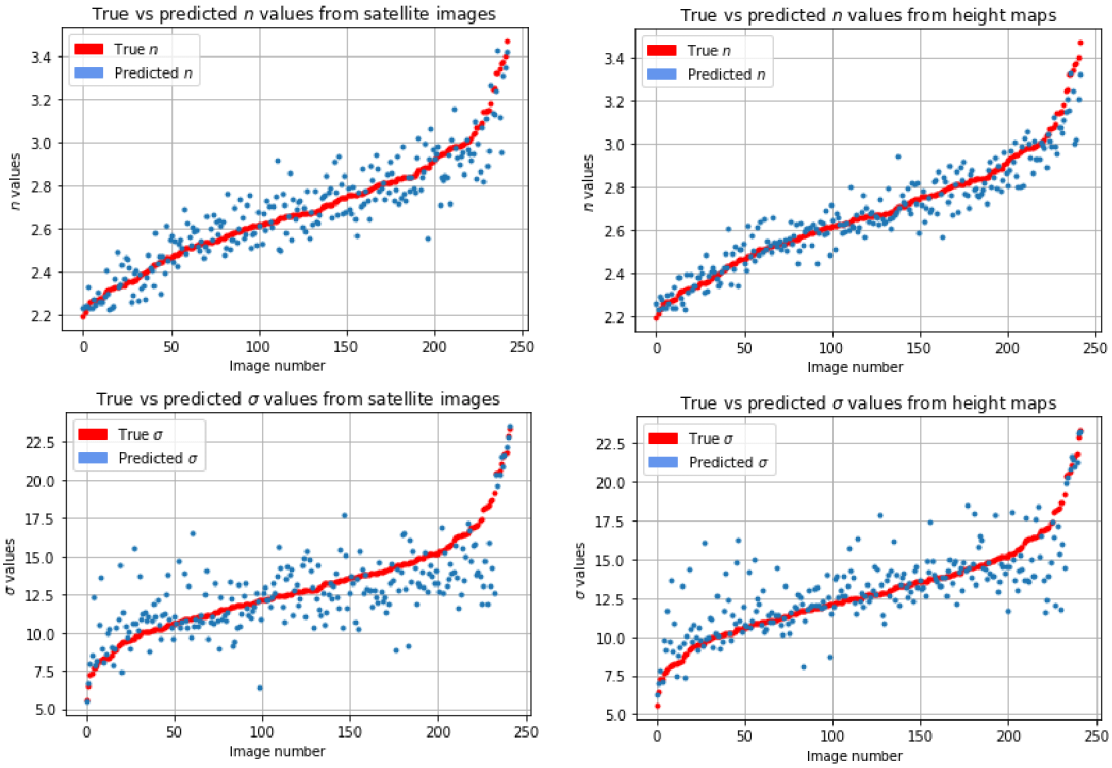
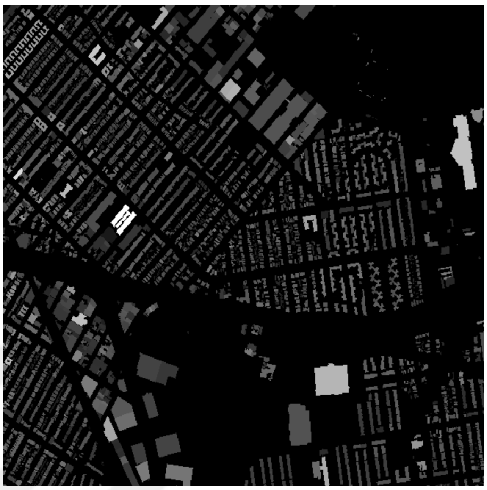


Figure 5.1.4. Prediction of test samples based on the low-complexity architecture [5].

(a) True n: 2.54
Predicted n: 2.52
True σ: 11.08
Predicted σ: 9.36



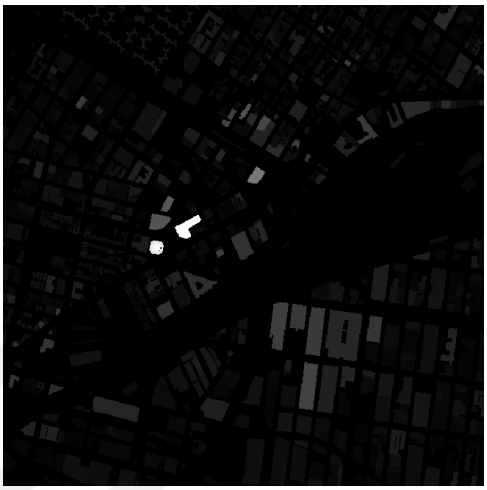
(b) True n: 2.54
Predicted n: 2.54
True σ: 11.08
Predicted σ: 10.47

(c) True n: 2.88
Predicted n: 2.89
True σ: 13.57
Predicted σ: 12.93

(d) True n: 2.88
Predicted n: 2.88
True σ: 13.57
Predicted σ: 13.43

Figure 5.1.5. Sample results from the test set.

(a) True n: 3.14
Predicted n: 3.12
True σ: 22.87
Predicted σ: 21.05



(b) True n: 3.14
Predicted n: 3.00
True σ: 22.87
Predicted σ: 21.21

Figure 5.1.6. Sample results from the test set.

5.2 Conditional GAN performance for excessive path loss prediction

Table 5.3: GAN Path Loss Prediction Using Height Map

Transmitter Height	**80 m**	**300 m**
Average probability distribution mse	3.45×10^{-5}	1.28×10^{-4}
Average point-wise mse	451	255
Path loss variance	520	277

In this work, the conditional GAN model is similar to style transferring because the model is taking a height map as an input and predicting the path loss image of that region. This method has been tried on two different transmitter altitudes which are 80 and 300 meters. The performance of the modified conditional generative adversarial network for excessive path loss prediction in Chapter 4 has been evaluated and results are given in Table 5.3.

The conditional GAN model is evaluated by average point-wise mean squared error and average probability distribution mean squared error. Average point-wise mean squared error is the evaluation parameter in which we compare all the pixels in the predicted and ground truth images excluding the indoor pixels. While for the average probability distribution mean squared error, we calculate across the region the probability distribution of ground truth and predicted path losses and we divide the distribution into 300-bins where each bin represents 1 dB and we calculate the mean squared error between both of them. When we look at figure 5.2.1 (b) the true path loss image, it can be seen that most of the area in between the buildings is covered in red which is indicated as high path loss values. It can be explained that since the transmitter height is set to 80 meters in the middle of the image, most of the buildings around the Tx are higher than 80 meters which can result in high path losses for the receivers. Since we are only interested in outdoor path loss modeling, the mean squared error between the ground truth of path loss image and predicted path loss image is calculated for the outdoor receivers for each of the test samples. In Table 5.3 the average point-wise mean squared error in 300 meters has a better result than 80 meters. We can see that for 300 meters and 80 meters datasets, average point-wise mse is less than the true path losses variance which means that model is not predicting the average but actually trying to give meaningful estimations.

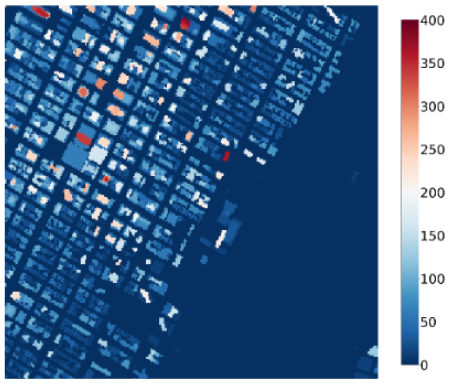
Even though the 80 meters path loss images has higher variance where we can see many images that have wide regions of high path losses that the network can learn from and successfully be able to predict the shadowing effect as we can notice the hint of shadowing in 5.2.1 (c). We think that with a bigger dataset that has many more interesting samples like in 5.2.1 the network would be able to estimate the shadowing more accurately.

It can be noticed from Figure 5.2.1 (d) that the hight path loss values are very concentrated in narrow-based high triangle, while the prediction is widely distributing this probability across the bigger interval and as we said that with a better balanced dataset this could reach the desired behaviour.

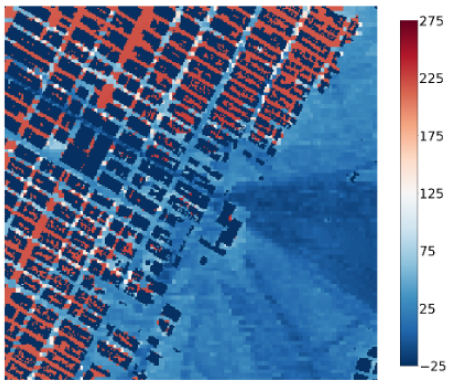
In a contrast to 80m, the 300 meters transmitter is too high for most of the

buildings so that the high path loss does not usual occur. That's why the performance is not as well as in the case of the 80 meters transmitter, higher variance and higher more variations.

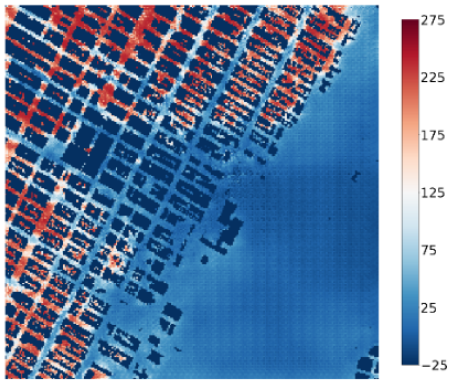
In order to show the performance of the proposed method, some of the results for 80 meters and 300 meters are demonstrated in Figure 5.2.3 and 5.2.4.



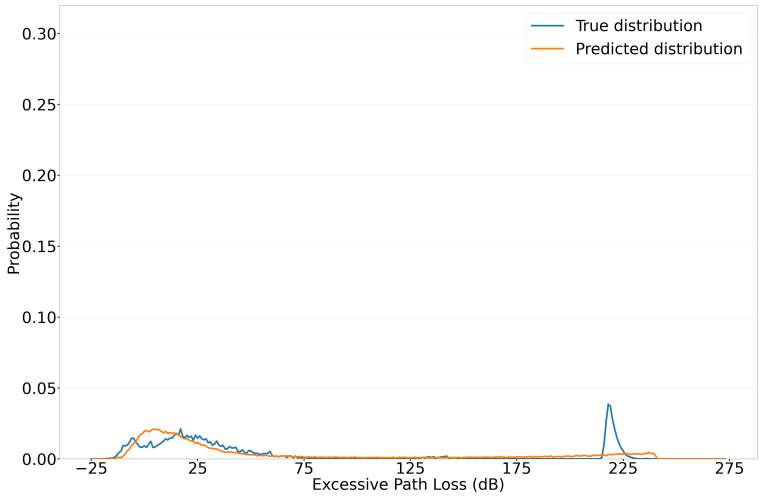
(a) Colored height map



(b) Ground truth path loss image



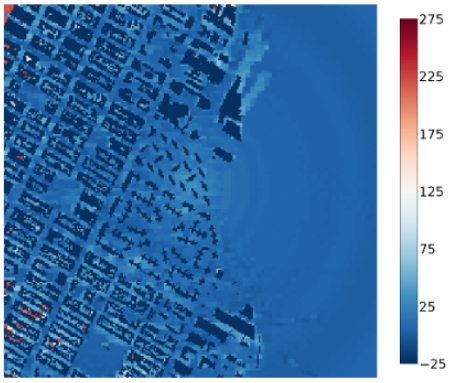
(c) Predicted path loss image



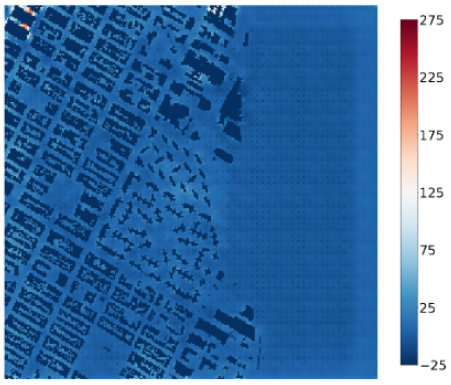
(d) Probability distribution of path losses

Figure 5.2.1: One of the 80 meter result is shown.

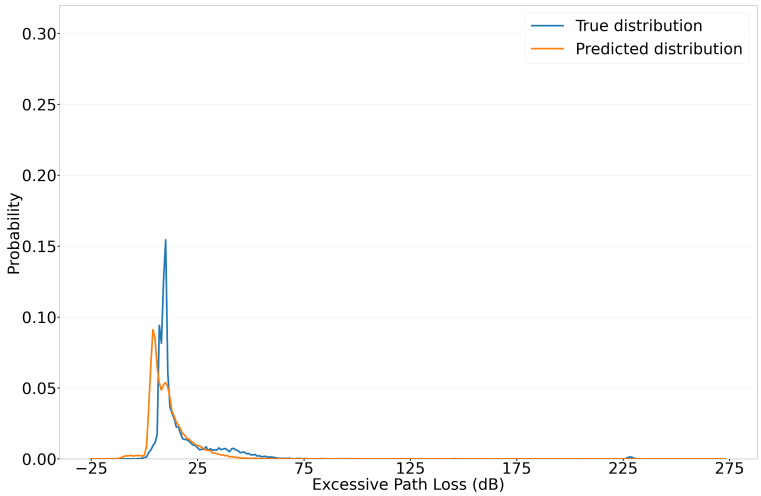
(a) Colored height map



(b) Ground truth path loss image



(c) Predicted path loss image



(d) Probability distribution of path losses

Figure 5.2.2: One of the 300 meter result is shown.

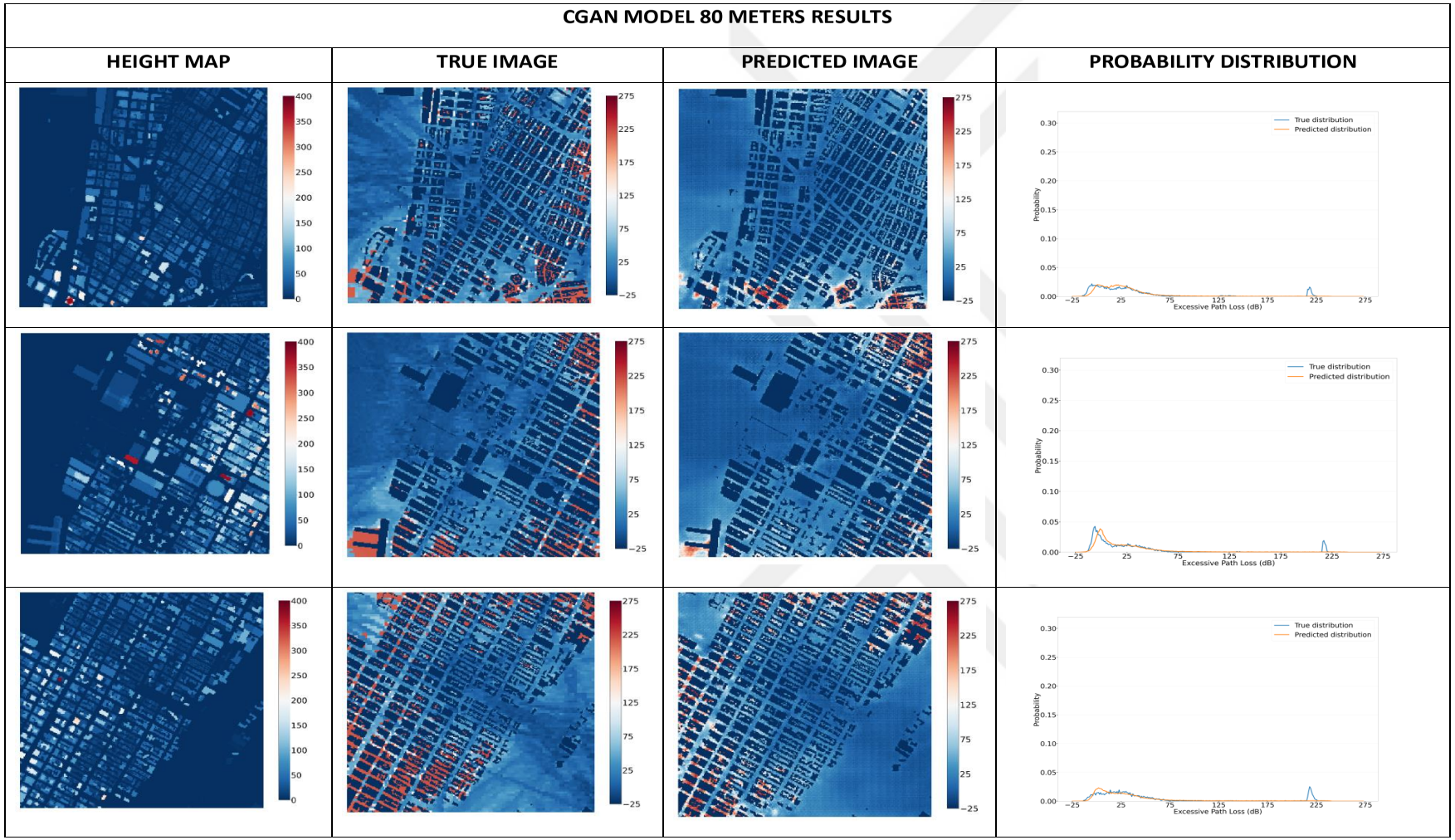


Figure 5.2.3: Some test sample results for 80 meter.

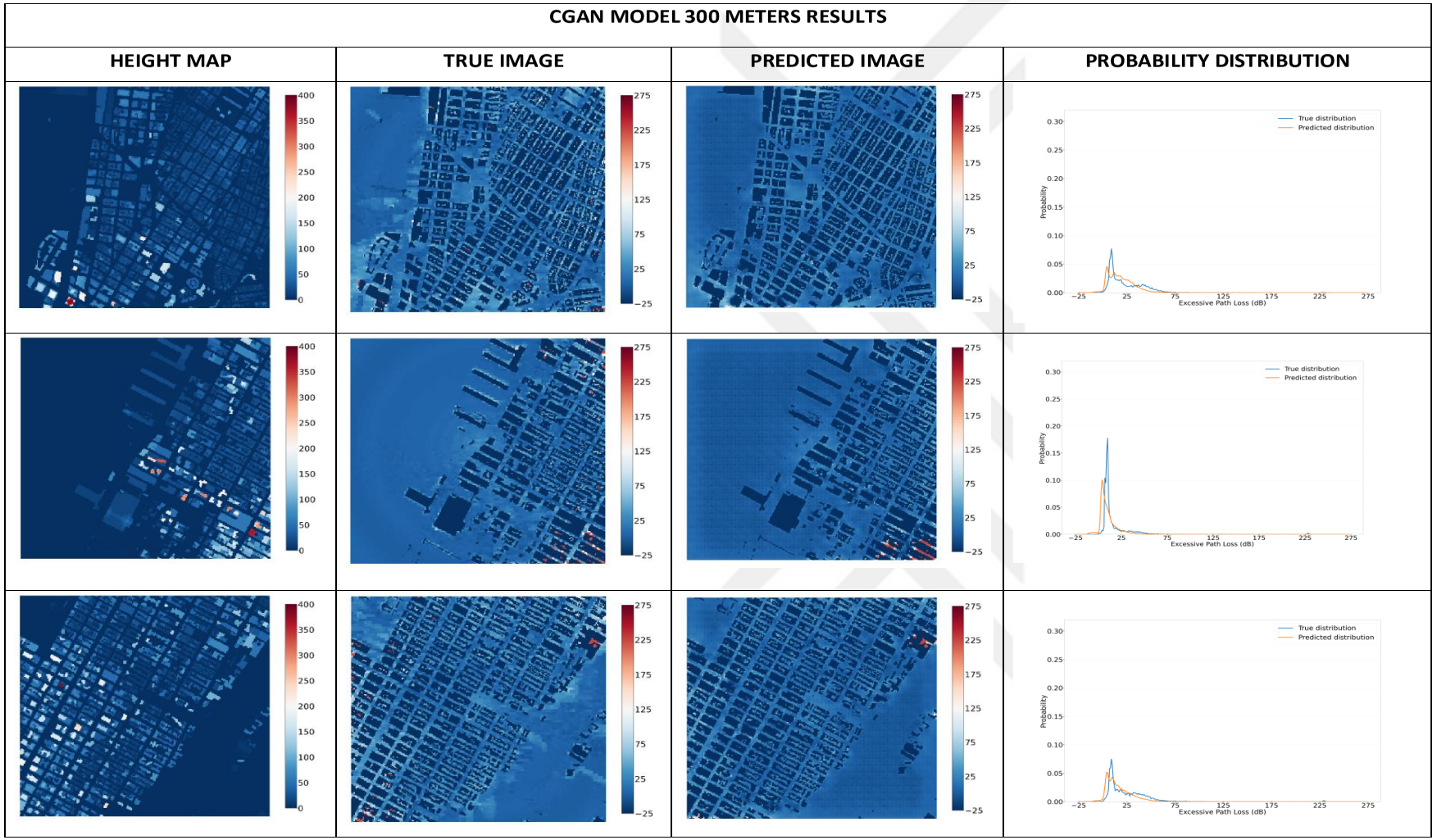


Figure 5.2.4: Some test sample results for 300 meter.

Chapter 6

Conclusion

In this study, two different problems are indicated on the wireless channel communication which are estimating the path loss parameters such as (n and σ) and estimating excessive path loss point-wisely using deep learning from height map. We show that whenever the height maps are available, it should be preferred as they incorporate the 3D structural information directly while structural information is extracted implicitly from satellite images through the networks. Our first problem was the modelling a regression of path loss exponent (n) and shadowing factor (σ) estimation that is issued by height map and satellite images and trained with two separate network architectures. It is observed that VGG-16 architecture is suitable for n estimation because the n value is not noisy, so it is less prone to over-fitting [42]. While in the σ case more noise is present, and VGG-16 learns the noise during training, which badly affects the testing performance. Hence, a shallower network is preferable to get rid of the over-fitting caused by VGG-16 in σ estimation. The performance of satellite images and height maps are also compared and height maps are shown to be better at estimating the channel parameters. The proposed approach can be easily adapted to different communication scenarios, with different transmitter frequencies and heights. Also, it remains to be seen whether better performance can be obtained by combining both 2D satellite image and height map information when training the deep network.

On the other hand which we surely think that a difficult task of using the conditional GAN for predicting the point-wise excessive path loss of a targeted area is taken as the second problem to work on. With the proposed method it can be said that conditional GAN is much more appropriate for the prediction of excessive path loss for a region and its benefits from the height info. Even when it is compared with [4], we achieved more realistic and accurate results and instead of predicting 10 dB wide-bin probability distribution, we are predicting more accurately 1-bin probability distribution. From that, it is proved that point-wise path loss prediction gives helpful information and improves the excessive path loss probability distribution prediction very well. Moreover, similar to these studies, the use of deep neural networks can be investigated in predicting other channel features, such as the delay spread function, which is important in the design and performance of wireless communication systems.

Consequently, these works can be used to get a more general network planning of a region and its instant needs. In real-life scenarios, this type of service can be used in a disaster zone, i.e. to increase communication in that region with the UAV, like how much power it should give, in which meters that transmitter needs to be stay or its optimization, such as channel parameters.

Bibliography

[1] E. Loosli, "Drone photogrammetry vs. lidar: what sensor to choose for a given application," *Wingtra*, Mar 2020.

[2] P. Medeđović, M. Veletić, and Ž. Blagojević, "Wireless insite software verification via analysis and comparison of simulation and measurement results," in *Proceedings of the 35th International Convention MIPRO*, pp. 776–781, IEEE, 2012.

[3] G. Yang, Y. Zhang, Z. He, J. Wen, Z. Ji, and Y. Li, "Machine-learning-based prediction methods for path loss and delay spread in air-to-ground millimetre-wave channels," *IET Microwaves, Antennas & Propagation*, vol. 13, no. 8, pp. 1113–1121, 2019.

[4] O. Ahmadien, H. F. Ates, T. Baykas, and B. K. Gunturk, "Predicting path loss distribution of an area from satellite images using deep learning," *IEEE Access*, 2020.

[5] J. Thrane, D. Zibar, and H. L. Christiansen, "Model-aided deep learning method for path loss prediction in mobile communication systems at 2.6 GHz," *IEEE Access*, vol. 8, pp. 7925–7936, 2020.

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134, 2017.

[7] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232, 2017.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[9] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[10] T. K. Sarkar, Z. Ji, K. Kim, A. Medouri, and M. Salazar-Palma, "A survey of various propagation models for mobile communication," *IEEE Antennas and Propagation Magazine*, vol. 45, no. 3, pp. 51–82, 2003.

[11] V. Abhayawardhana, I. Wassell, D. Crosby, M. Sellars, and M. Brown, "Comparison of empirical propagation path loss models for fixed wireless access systems," in *IEEE 61st Vehicular Technology Conference*, vol. 1, pp. 73–77, IEEE, 2005.

[12] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE Transactions on Vehicular Technology*, vol. 29, no. 3, pp. 317–325, 1980.

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

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, Ieee, 1998.

[16] H. F. Ates, S. M. Hashir, T. Baykas, and B. K. Gunturk, "Path loss exponent and shadowing factor prediction from satellite images using deep learning," *IEEE Access*, vol. 7, pp. 101366–101375, 2019.

[17] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[18] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," in *Proceedings of the IEEE*, vol. 109, pp. 43–76, IEEE, 2020.

[19] S. Srinivasa and M. Haenggi, "Path loss exponent estimation in large wireless networks," in *Information Theory and Applications Workshop*, pp. 124–129, IEEE, 2009.

[20] Y. Singh, "Comparison of okumura, hata and cost-231 models on the basis of path loss and signal strength," *International Journal of Computer Applications*, vol. 59, no. 11, 2012.

[21] D. J. Cichon and T. Kürner, "Propagation prediction models," *COST-231 TD*, vol. 95, no. 66, pp. 115–207, 1995.

[22] 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 Journal on Selected Areas in Communications*, vol. 35, pp. 1678–1690, July 2017.

[23] M. Zhu, A. Singh, and F. Tufvesson, "Measurement based ray launching for analysis of outdoor propagation," in *6th European Conference on Antennas and Propagation (EUCAP)*, pp. 3332–3336, March 2012.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

[25] G. Biau, B. Cadre, M. Sangnier, U. Tanielian, *et al.*, "Some theoretical properties of gans," *Annals of Statistics*, vol. 48, no. 3, pp. 1539–1566, 2020.

[26] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5485–5493, 2017.

[27] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4681–4690, 2017.

[28] S. Yang, L. Xie, X. Chen, X. Lou, X. Zhu, D. Huang, and H. Li, "Statistical parametric speech synthesis using generative adversarial networks under a multi-task learning framework," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 685–691, IEEE, 2017.

[29] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[30] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[31] Y. Zhang, J. Wen, G. Yang, Z. He, and J. Wang, "Path loss prediction based on machine learning: Principle, method, and data expansion," *Applied Sciences*, vol. 9, no. 1908, pp. 1–18, 2019.

[32] M. Neunerdt, A. Engels, and R. Mathar, "Land use classification as a key component for path loss prediction in rural areas," in *7th International Symposium on Wireless Communication Systems*, pp. 666–670, Sep. 2010.

[33] 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 Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 576–584, 2017.

[34] 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 Communications and Mobile Computing*, vol. 2018, 2018.

[35] N. Bitar, S. Muhammad, and H. H. Refai, "Wireless technology identification using deep convolutional neural networks," in *IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–6, Oct 2017.

[36] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*, vol. 2. Prentice Hall PTR New Jersey, 1996.

[37] 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 Communications Letters*, vol. 6, no. 4, pp. 514–517, 2017.

[38] K. Kelchtermans and T. Tuytelaars, "How hard is it to cross the room?–training (recurrent) neural networks to steer a UAV," *arXiv preprint arXiv:1702.07600*, 2017.

[39] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017.

[40] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *IEEE Global Communications Conference*, pp. 2898–2904, IEEE, 2014.

[41] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *European Conference on Computer Vision*, pp. 702–716, Springer, 2016.

[42] D. M. Hawkins, "The problem of overfitting," *Journal of Chemical Information and Computer Sciences*, vol. 44, no. 1, pp. 1–12, 2004.

Appendix A

Summaries of the Networks

A.1 Summary of VGG-16 network

Layer (type)	Output Shape	Param #
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
dense_1 (Dense)	(None, 1)	4097

Total params: 134,264,641
Trainable params: 134,264,641
Non-trainable params: 0

A.2 Summary of low-complexity network

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
zero_padding2d_1 (ZeroPaddin	(None, 228, 228, 3)	0
conv2d_1 (Conv2D)	(None, 224, 224, 200)	15200
activation_1 (Activation)	(None, 224, 224, 200)	0
batch_normalization_1 (Batch	(None, 224, 224, 200)	800
max_pooling2d_1 (MaxPooling2	(None, 112, 112, 200)	0
zero_padding2d_2 (ZeroPaddin	(None, 116, 116, 200)	0
conv2d_2 (Conv2D)	(None, 114, 114, 100)	180100
activation_2 (Activation)	(None, 114, 114, 100)	0
batch_normalization_2 (Batch	(None, 114, 114, 100)	400
max_pooling2d_2 (MaxPooling2	(None, 57, 57, 100)	0
zero_padding2d_3 (ZeroPaddin	(None, 61, 61, 100)	0
conv2d_3 (Conv2D)	(None, 59, 59, 50)	45050
activation_3 (Activation)	(None, 59, 59, 50)	0
batch_normalization_3 (Batch	(None, 59, 59, 50)	200
max_pooling2d_3 (MaxPooling2	(None, 29, 29, 50)	0
zero_padding2d_4 (ZeroPaddin	(None, 33, 33, 50)	0
conv2d_4 (Conv2D)	(None, 31, 31, 25)	11275
activation_4 (Activation)	(None, 31, 31, 25)	0
batch_normalization_4 (Batch	(None, 31, 31, 25)	100
max_pooling2d_4 (MaxPooling2	(None, 15, 15, 25)	0

zero_padding2d_5 (ZeroPaddin	(None, 19, 19, 25)	0
conv2d_5 (Conv2D)	(None, 18, 18, 12)	1212
activation_5 (Activation)	(None, 18, 18, 12)	0
batch_normalization_5 (Batch	(None, 18, 18, 12)	48
max_pooling2d_5 (MaxPooling2	(None, 9, 9, 12)	0
zero_padding2d_6 (ZeroPaddin	(None, 13, 13, 12)	0
conv2d_6 (Conv2D)	(None, 12, 12, 1)	49
activation_6 (Activation)	(None, 12, 12, 1)	0
batch_normalization_6 (Batch	(None, 12, 12, 1)	4
max_pooling2d_6 (MaxPooling2	(None, 6, 6, 1)	0
flatten_1 (Flatten)	(None, 36)	0
dense_1 (Dense)	(None, 200)	7400
activation_7 (Activation)	(None, 200)	0
dense_2 (Dense)	(None, 16)	3216
activation_8 (Activation)	(None, 16)	0
dense_3 (Dense)	(None, 1)	17
activation_9 (Activation)	(None, 1)	0

Total params: 265,071
Trainable params: 264,295
Non-trainable params: 776

A.3 Summary of GAN generator summary

```
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 256, 256, 3)      0
_________________________________________________________________
sequential_6 (Sequential)    (None, 128, 128, 64)      3072
_________________________________________________________________
sequential_7 (Sequential)    (None, 64, 64, 128)       131584
_________________________________________________________________
sequential_8 (Sequential)    (None, 32, 32, 256)       525312
_________________________________________________________________
sequential_9 (Sequential)    (None, 16, 16, 512)       2099200
_________________________________________________________________
sequential_10 (Sequential)   (None, 8, 8, 512)         4196352
_________________________________________________________________
sequential_11 (Sequential)   (None, 4, 4, 256)         2098176
_________________________________________________________________
sequential_12 (Sequential)   (None, 2, 2, 256)         1049600
_________________________________________________________________
sequential_13 (Sequential)   (None, 1, 1, 256)         1049600
_________________________________________________________________
sequential_14 (Sequential)   (None, 2, 2, 256)         1049600
_________________________________________________________________
concatenate (Concatenate)    (None, 2, 2, 512)         0

_________________________________________________________________
sequential_15 (Sequential)   (None, 4, 4, 256)         2098176
_________________________________________________________________
concatenate_1 (Concatenate)  (None, 4, 4, 512)         0

_________________________________________________________________
sequential_16 (Sequential)   (None, 8, 8, 256)         2098176
_________________________________________________________________
concatenate_2 (Concatenate)  (None, 8, 8, 768)         0

_________________________________________________________________
sequential_17 (Sequential)   (None, 16, 16, 256)       3146752
_________________________________________________________________
concatenate_3 (Concatenate)  (None, 16, 16, 768)       0

_________________________________________________________________
sequential_18 (Sequential)   (None, 32, 32, 128)       1573376
_________________________________________________________________
concatenate_4 (Concatenate)  (None, 32, 32, 384)       0

_________________________________________________________________
sequential_19 (Sequential)   (None, 64, 64, 64)        393472
_________________________________________________________________
concatenate_5 (Concatenate)  (None, 64, 64, 192)       0

_________________________________________________________________
sequential_20 (Sequential)   (None, 128, 128, 32)      98432
_________________________________________________________________
concatenate_6 (Concatenate)  (None, 128, 128, 96)      0

_________________________________________________________________
conv2d_transpose_9 (Conv2DTrans (None, 256, 256, 3)    4611
=================================================================
Total params: 21,615,491
Trainable params: 21,608,643
Non-trainable params: 6,848
```

A.4 Summary of GAN discriminator summary

Layer (type)	Output Shape	Param #
input_image (InputLayer)	[(None, 256, 256, 3)	0
target_image (InputLayer)	[(None, 256, 256, 3)	0
concatenate_7 (Concatenate)	(None, 256, 256, 6)	0
sequential_21 (Sequential)	(None, 128, 128, 64)	6144
sequential_22 (Sequential)	(None, 64, 64, 128)	131584
sequential_23 (Sequential)	(None, 32, 32, 256)	525312
conv2d_15 (Conv2D)	(None, 29, 29, 512)	2097152
batch_normalization_22 (BatchNo	(None, 29, 29, 512)	2048
leaky_re_lu_15 (LeakyReLU)	(None, 29, 29, 512)	0
conv2d_16 (Conv2D)	(None, 26, 26, 256)	2097152
batch_normalization_23 (BatchNo	(None, 26, 26, 256)	1024
leaky_re_lu_16 (LeakyReLU)	(None, 26, 26, 256)	0
conv2d_17 (Conv2D)	(None, 23, 23, 128)	524288
batch_normalization_24 (BatchNo	(None, 23, 23, 128)	512
leaky_re_lu_17 (LeakyReLU)	(None, 23, 23, 128)	0
conv2d_18 (Conv2D)	(None, 20, 20, 64)	131072
batch_normalization_25 (BatchNo	(None, 20, 20, 64)	256
leaky_re_lu_18 (LeakyReLU)	(None, 20, 20, 64)	0
conv2d_19 (Conv2D)	(None, 17, 17, 32)	32768
batch_normalization_26 (BatchNo	(None, 17, 17, 32)	128
leaky_re_lu_19 (LeakyReLU)	(None, 17, 17, 32)	0

```
_________________________________________________________________
conv2d_20 (Conv2D)              (None, 14, 14, 32)   16384
_________________________________________________________________
batch_normalization_27 (BatchNo (None, 14, 14, 32)   128
_________________________________________________________________
leaky_re_lu_20 (LeakyReLU)      (None, 14, 14, 32)   0
_________________________________________________________________
conv2d_21 (Conv2D)              (None, 11, 11, 32)   16384
_________________________________________________________________
batch_normalization_28 (BatchNo (None, 11, 11, 32)   128
_________________________________________________________________
leaky_re_lu_21 (LeakyReLU)      (None, 11, 11, 32)   0
_________________________________________________________________
conv2d_22 (Conv2D)              (None, 8, 8, 32)     16384
_________________________________________________________________
batch_normalization_29 (BatchNo (None, 8, 8, 32)     128
_________________________________________________________________
leaky_re_lu_22 (LeakyReLU)      (None, 8, 8, 32)     0
_________________________________________________________________
conv2d_23 (Conv2D)              (None, 5, 5, 1)      513
=================================================================
Total params: 5,599,489
Trainable params: 5,596,545
Non-trainable params: 2,944
```

PATH LOSS PREDICTION FROM HEIGHT MAP USING DEEP LEARNING

ORIGINALITY REPORT

10%	6%	7%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Eastern Mediterranean University Student Paper	1%
2	www.institut-fuer-menschenrechte.de Internet Source	1%
3	Omar Ahmadien, Hasan F. Ates, Tuncer Baykas, Bahadir K. Gunturk. "Predicting Path Loss Distribution of an Area From Satellite Images Using Deep Learning", IEEE Access, 2020 Publication	1%
4	Submitted to Nashville State Community College Student Paper	1%
5	Guanshu Yang, Yan Zhang, Zunwen He, Jinxiao Wen, Zijie Ji, Yue Li. "Machine-learning-based prediction methods for path loss and delay spread in air-to-ground millimetre-wave channels", IET Microwaves, Antennas & Propagation, 2019	<1%